

An Ensemble Learning Approach for Improved Loan Fraud Detection: Comparing and Combining Machine Learning Models



Anuradha Anuradha

Student No: 10634610

Dublin Business School

Applied Research Project submitted in partial fulfilment of the requirements of

MSc. in Data Analytics

at Dublin Business School

Supervisor: Dr. Samuel Ogwu

January 2024

Declaration

‘I declare that this Applied Research Project I have submitted to Dublin Business School for the award of MSc. Data Analytics is the result of my own investigations. Except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.’

Signed: ANURADHA ANURADHA

Student Number: 10634610

Date: 08th January 2024.

Acknowledgments

I want to express my heartfelt gratitude for the invaluable guidance I received from my supervisor, Dr. Samuel Ogwu. I am profoundly grateful for his constant support, which has been instrumental in making this endeavour possible and his willingness to give his time generously has been very much appreciated.

I would like to thank my Professors Ms. Terri Hoare, Amit Sharma, Kunwar Madan, Courtney Ford, and Shahram Azizi Sazi for their significant contributions to my academic growth. The combined efforts of all the professors have played a pivotal role in shaping my abilities as I maneuverer through the challenges of my MSc. In Data Analytics.

Table of Contents

Abstract	2
Chapter 1 Introduction	13
1.1 Background and Significance	13
1.2 Motivation.....	14
1.3 Problem Statement.....	15
1.4 Research Questions.....	16
1.5 Research Hypothesis.....	17
1.6 Research Objectives.....	17
1.7 Research Gaps.....	17
Chapter 2 Literature Review.....	18
2.1 Research studies employing Long Short-Term Memory (LSTM) model	18
2.2 Research studies employing Convolutional Neural Network (CNN).....	21
2.3 Research studies employing the AdaBoost and Random Forest model.....	24
2.4 Summary of Literature Review.....	28
Chapter 3 Research Methodology.....	29
3.1 Methodology	30

	4
3.1.1 Business Understanding.....	31
3.1.2 Data Understanding	32
3.1.3 Data Preparation.....	32
3.1.4 Modelling.....	33
3.2 Data Visualisation	33
3.3 Data Description	42
3.4 Data Collection and Analysis.....	42
3.5 Feature Selection.....	43
3.5.1 Forward Feature Selection.....	44
3.5.2 Reverse (Backward) Feature Selection.....	44
3.5.3 Automatic Feature Selection.....	44
3.6 List of Models.....	45
3.6.1 Logistic Regression for PPP Loan Fraud Detection	45
3.6.2 Random Forest for PPP Loan Fraud Detection.....	45
3.6.3 AdaBoost for PPP Loan Fraud Detection	46
3.6.4 LSTM (Long Short-Term Memory) Networks for PPP Loan Fraud Detection.....	46
3.6.5 CNN (Convolutional Neural Network) for PPP Loan Fraud Detection	46

Chapter 4 Results and Analysis.....	47
4.1 Results without Feature Selection.....	47
4.1.1 Logistic Regression.....	47
4.1.2 Random Forest.....	48
4.1.3 AdaBoost.....	49
4.1.4 LSTM.....	49
4.1.5 CNN.....	50
4.2 Results with Forward Feature Selection.....	51
4.2.1 Logistic Regression.....	51
4.2.2 Random Forest.....	53
4.2.3 AdaBoost.....	54
4.2.4 LSTM.....	55
4.2.5 CNN.....	56
4.3 Results with Backward Feature Selection.....	58
4.3.1 Logistic Regression.....	58
4.3.2 Random Forest.....	59
4.3.3 AdaBoost.....	60

	6
4.3.4 LSTM.....	61
4.3.5 CNN.....	62
4.4 Results with Automatic Feature Selection.....	64
4.4.1 Logistic Regression.....	64
4.4.2 Random Forest.....	65
4.4.3 AdaBoost.....	66
4.4.4 LSTM.....	67
4.4.5 CNN.....	68
4.5 Comparison of the Models.....	70
Chapter 5 Discussion.....	71
Chapter 6 Conclusion, Limitations and Future Scope.....	75
References.....	77
Appendix.....	81

Table of Figures

Figure 3.1: Methodology Flow	30
Figure 3.2: CRISP-DM Methodology.....	31
Figure 3.3: Processing Method vs Count of Applications with respect to Loan Status.....	34
Figure 3.4: LMIIndicator vs Count of Applications with respect to Loan Status.....	34
Figure 3.5: Hubzone Indicator vs Count of Applications with respect to Loan Status.....	35
Figure 3.6: Number of Projects with respect to the state.....	36
Figure 3.7: Number of borrowers vs count of Applications per state.....	36
Figure 3.8: Class distribution.....	37
Figure 3.9: Ethnicity distribution in the dataset.....	38
Figure 3.10: Business type vs count of Applications.....	38
Figure 3.11: KDE plot for Total Proceed.....	39
Figure 3.12: KDE plot for Initial Approval Amount.....	39
Figure 3.13: KDE plot for Current Approval Amount.....	40
Figure 3.14: Correlation matrix for dataset features.....	41
Figure 4.1: ROC for Logistic Regression with Forward Feature Selection.....	52
Figure 4.2: ROC for Random Forest with Forward Feature Selection.....	53

Figure 4.3: ROC for AdaBoost with Forward Feature Selection.....	55
Figure 4.4: ROC for LSTM model with Forward Feature Selection.....	56
Figure 4.5: ROC for CNN with Forward Feature Selection.....	57
Figure 4.6: ROC for Logistic Regression with Backward Feature Selection.....	59
Figure 4.7: ROC for Random Forest with Backward Feature Selection.....	60
Figure 4.8: ROC for AdaBoost with Backward Feature Selection.....	61
Figure 4.9: ROC for LSTM with Backward Feature Selection.....	62
Figure 4.10: ROC for CNN with Backward Feature Selection.....	63
Figure 4.11: ROC for Logistic Regression with Automatic Feature Selection.....	64
Figure 4.12: ROC for Random Forest with Automatic Feature Selection.....	66
Figure 4.13: ROC for AdaBoost with Automatic Feature Selection.....	67
Figure 4.14: ROC for LSTM with Automatic Feature Selection.....	68
Figure 4.15: ROC for CNN with Automatic Feature Selection.....	69

Table of Tables

Table 4.1: Classification report and Confusion Matrix for the Logistic Regression model	47
Table 4.2: Classification report and Confusion Matrix for the Random Forest model.....	48
Table 4.3: Classification report and Confusion Matrix for the AdaBoost model	49
Table 4.4: Classification report and Confusion Matrix for the LSTM model.....	50
Table 4.5: Classification report and Confusion Matrix for the CNN model.....	51
Table 4.6: Performance of Logistic Regression model with Forward Feature Selection	52
Table 4.7: Performance of Random Forest model with Forward Feature Selection.....	53
Table 4.8: Performance of AdaBoost model with Forward Feature Selection.....	54
Table 4.9: Performance of LSTM model with Forward Feature Selection.....	55
Table 4.10: Performance of CNN model with Forward Feature Selection.....	57
Table 4.11: Performance of Logistic Regression model with Backward Selection.....	58
Table 4.12: Performance of Random Forest model with Backward Selection	59
Table 4.13: Performance of AdaBoost model with Backward Selection.....	60
Table 4.14: Performance of LSTM model with Backward Selection.....	61
Table 4.15: Performance of CNN model with Backward Selection	63
Table 4.16: Performance of Logistic Regression model with Automatic Selection	64

Table 4.17: Performance of Random Forest model with Automatic Selection.....	65
Table 4.18: Performance of AdaBoost model with Automatic Selection	66
Table 4.19: Performance of LSTM model with Automatic Selection.....	67
Table 4.20: Performance of CNN model with Automatic Selection.....	68
Table 5.1: Comparison of the Models.....	70

Table of Acronyms

LSTM	Long Short-Term Memory
CNN	Convolutional Neural Networks
AdaBoost	Adaptive Boosting
PPP	Paycheck Protection Program
CRISP-DM	Cross Industry Standard Process for Data Mining
EDA	Exploratory Data Analysis
SVM	Support Vector Machine
GA	Genetic Algorithm
RFE	Recursive Feature Elimination
DFS	Deep Feature Synthesis
GAN	Generative Adversarial Networks
SMOTE	Synthetic Minority Oversampling Technique
MLP	Multi-layer Perceptron
AUC	Area Under the Curve
KNN	K-Nearest Neighbors
NLP	Natural Language Processing
RMSE	Root Mean Square Error
XAI	Explainable Artificial Intelligence
SC	Smart Contracts
RNN	Recurrent Neural Network
LMI	Low-to-Moderate Income
XGBoost	Extreme Gradient Boosting
MCC	Matthews Correlation Coefficient
FNN	Feed-forward Neural Network

Abstract

This thesis investigated loan fraud detection using advanced machine learning techniques, focusing on Logistic Regression, Random Forest, AdaBoost, Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNN). The study emphasized the importance of feature selection, and explored forward, backward, and automatic methods to improve model performance. Comparative analysis across models revealed that Random Forest consistently outperforms other models in accuracy and efficiency, regardless of the feature selection technique. AdaBoost showed consistent results but at a higher computational cost, while LSTM and CNN were highly sensitive to the choice of feature selection, affecting their performance significantly. The thesis concluded that feature selection was vital for optimizing machine learning models for fraud detection, with the impact varying significantly across different algorithms. Random Forest emerged as a robust and efficient model for fraud detection, adaptable to various applications. The findings underscored the potential of machine learning to strengthen financial security and trust.

Chapter 1 Introduction

The financial sector is increasingly turning to advanced machine learning techniques to combat the pervasive and evolving challenge of loan fraud. This thesis explores an innovative ensemble learning approach, using Logistic Regression, Random Forest, Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Adaptive Boosting (AdaBoost) to enhance the detection and prevention of fraudulent activities. While the primary focus is on finding a robust ensemble model, an equally important aspect of this research is the identification and optimization of features that significantly influence the model's performance.

1.1 Background and Significance

The financial sector is increasingly grappling with the challenge of loan fraud, a critical issue that not only leads to substantial financial losses but also undermines customer trust. In the rapidly evolving landscape of global finance, characterized by advanced technology, novel financial products, and fast-paced international transactions, fraudsters continuously adapt and refine their strategies. This dynamic environment makes the detection and prevention of loan fraud both costly and complex for financial institutions. Traditional statistical methods and standalone machine learning models have shown some effectiveness, but they often fall short in adapting to the sophisticated and evolving nature of fraudulent activities.

In this scenario, ensemble learning approaches have gained prominence as a potent strategy to enhance the accuracy and reliability of loan fraud detection systems. These methods combine the strengths of various machine learning models, such as Random Forest, Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN), to form a more

robust and precise predictive model. This study focuses on identifying an ensemble model such as Random Forest, LSTM, CNN, AdaBoost models, to detect intricate and sophisticated patterns of loan fraud and significant features that may be missed by individual models. The challenge lies in selecting appropriate ensemble techniques and configuring them optimally to find significant features of loan fraud dataset, an area that remains ripe for exploration and innovation.

1.2 Motivation

The motivation for this research is deeply rooted in the urgent need to tackle the escalating challenges of loan fraud in the financial sector, a need driven by several critical factors. Firstly, the evolving nature of fraud, with financial fraudsters continually adapting their tactics, necessitates more advanced and dynamic methods to counteract these threats. Traditional models, effective in the past, are increasingly inadequate against sophisticated and ever-changing fraudulent schemes. Secondly, rapid advancements in machine learning, particularly in Random Forest, LSTM and CNN, present a unique opportunity to revolutionize fraud detection. These technologies, known for their exceptional capabilities in feature importance ranking, pattern recognition and data analysis, have untapped potential in loan fraud detection.

Furthermore, the integration of models is not merely about enhancing individual models but creating a synergistic effect where combined strengths address individual weaknesses. This integrated approach is expected to yield a robust and versatile system, capable of detecting a wider range of fraudulent activities with greater accuracy. Another driving factor is the challenge of class imbalance in loan fraud datasets, where legitimate transactions often vastly outnumber fraudulent ones. An ensemble approach, especially one that adapts and learns from its errors like

Random Forest, LSTM, CNN and AdaBoost, can offer a more effective way to handle this imbalance, leading to more reliable detection outcomes.

The real-world impact of improving loan fraud detection extends beyond technical achievements. Effective fraud detection systems can significantly reduce financial losses, enhance customer trust, and contribute to the overall stability of the financial sector, crucial in an era where digital financial transactions are becoming the norm.

1.3 Problem Statement

The problem of loan fraud in the financial sector is a complex and ever-evolving challenge that significantly impacts institutions and stakeholders alike. At the core of this issue is the inadequacy of traditional fraud detection methods, which struggle to keep pace with the increasingly sophisticated techniques employed by fraudsters. These conventional methods often lack the necessary flexibility and advanced analytical capabilities required for effective detection and prevention of complex fraudulent activities. As a result, financial institutions face not only substantial financial losses but also a severe erosion of customer trust and confidence in their systems. This erosion of trust can have far-reaching consequences, potentially leading to a decrease in customer engagement and a tarnishing of the institution's reputation.

In response to these challenges, the integration of advanced machine learning techniques such as Random Forest, Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Adaptive Boosting (AdaBoost) as an ensemble learning model presents a promising solution. This approach aims to leverage the unique strengths of each model to create a more robust and accurate system for detecting loan fraud. LSTM networks, with their ability to

understand and predict sequences, are particularly adept at identifying patterns over time, which is crucial in recognizing fraudulent activities that unfold in a temporal sequence. CNNs, known for their pattern recognition capabilities, can effectively analyze structured data, such as transaction sequences, to uncover potential fraud. AdaBoost, as a meta-algorithm, enhances the performance of these models by focusing on the instances that are most difficult to classify, thereby improving the overall predictive power of the system.

However, the journey towards an effective ensemble learning model for loan fraud detection is fraught with challenges. One of the primary concerns is the complexity involved in integrating these diverse machine learning models. Each model has its own set of parameters and characteristics, and finding the optimal way to combine them requires careful tuning and a deep understanding of their interactions. Another significant challenge is the issue of class imbalance commonly found in loan fraud datasets. Fraudulent transactions are typically much less frequent than legitimate ones, which can lead to models that are biased towards predicting the majority class and thus failing to identify fraudulent activities effectively.

1.4 Research Questions

The research aimed to address the following questions:

1. Does the ensemble learning methods predict loan fraud better than traditional machine learning techniques?
2. Are feature selection techniques more likely to improve the accuracy and reliability of loan fraud prediction models?

1.5 Research Hypothesis

The hypotheses for the above research questions are mentioned below:

1. The Ensemble Model are more likely to outperform than traditional machine learning algorithms in the context of loan fraud prediction.
2. Feature selection techniques will increase the accuracy of loan fraud prediction models, resulting in more effective fraud prediction.

1.6 Research Objectives

The objectives of this research are as follows:

1. To compare the effectiveness of the ensemble learning model against traditional fraud prediction methods.
2. To investigate the influence of feature selection techniques on the accuracy and reliability of loan fraud prediction models, focusing on their contribution to fraud prediction.

1.7 Research Gaps

While ensemble learning has shown promise in various domains, its application in loan fraud detection, particularly using a Random Forest, LSTM, CNN, and AdaBoost, is not extensively explored. There is a gap in understanding how these specific models can be effectively used to identify the significant features and the extent to which they enhance fraud detection capabilities. Additionally, the practical implementation of such an ensemble model in the complex and regulated environment of financial institutions presents unexplored challenges. This research aims to fill these gaps by not only identifying an ensemble model but also by providing insights into its practical deployment and effectiveness in real-world scenarios.

Chapter 2 Literature Review

This section discusses a comprehensive inspection of research related to Loan Fraud detection using machine learning. It explores the methodologies employed in different studies pertaining to loan fraud detection.

2.1 Research studies employing Long Short-Term Memory (LSTM) model

The paper by Wang and Ni (2020) focuses on predicting the default risk in the peer-to-peer (P2P) lending market using an LSTM model that incorporates macroeconomic factors, specifically the unemployment rate. Their study is significant because it shifts from individual-level modelling to an aggregate level analysis, providing a comprehensive view of the P2P market. The research marks the first attempt to apply LSTM to aggregated sequence data in P2P lending, demonstrating its superiority over traditional time series models. It is also the first to include a macroeconomic factor, the unemployment rate, in LSTM modelling for repayment prediction at an aggregate level. The inclusion of this factor improved model performance.

The authors used Lending Club data, focusing on the monthly trend of default rates from 2007 to 2016. Their approach involved data preprocessing, including the removal of redundant information and aggregation of data by month. The LSTM model's performance was enhanced by incorporating the unemployment rate, resulting in lower root mean square error (RMSE) values on both training and testing datasets compared to traditional time series models. This study broadens the application of LSTM in the P2P market and offers valuable insights for investors, particularly in understanding the monthly trend of the default rate at an aggregate level.

The paper by Li *et. al.* (2018) presents a novel approach to predicting overdue bank loans. This paper aims to improve the accuracy of traditional user loan risk prediction models. The authors propose a model combining LSTM (Long Short-Term Memory) and SVM (Support Vector Machine) algorithms. LSTM is used to analyse the dynamic behaviour of users, while SVM focuses on users' static data. The data used includes users' basic information, bank records, browsing behaviour, credit card billing records, and loan times. The LSTM-SVM model shows a substantial improvement over traditional algorithms in predicting loan delinquency. The model effectively combines dynamic (user transactions) and static (user profile) data, resulting in high accuracy in overdue predictions.

The paper details the preprocessing of various data dimensions and the design of the LSTM-SVM algorithm. The model is trained on a significant dataset and shows a 99% accuracy rate in prediction, outperforming other traditional models like Bayesian, kNN, and standalone SVM or LSTM models. This study demonstrates the potential of combining LSTM and SVM for complex prediction tasks in the financial sector. However, the authors acknowledge limitations due to the dataset's privacy constraints and suggest future research to improve the model further.

Owolafe *et. al.* (2021) presents an LSTM-based model for credit card fraud detection with an aim to develop a Long Short-Term Memory-Recurrent Neural Network (LSTM-RNN) model to classify financial transactions as fraudulent or not. The research uses two datasets from Kaggle, applying Principal Component Analysis for feature extraction and Min-Max scalar algorithm for normalization. The LSTM-RNN model achieved a high classification accuracy of 99.58%, with a precision of 99.6% and a recall of 80%. The model demonstrated a significant reduction in the false alarm rate compared to previous systems.

The paper by Mohmad (2022) focuses on using LSTM for credit card fraud detection. The main aim is to assist bank management in scoring credit card clients using machine learning techniques, specifically using a bidirectional Long-Short Term Memory (LSTM) model. This model predicts the probability of missed payments for credit card customers and is trained on real credit card data. The LSTM model's performance is compared with four traditional machine learning algorithms: Support Vector Machine, Random Forest, Multi-Layer Perceptron Neural Network, and Logistic Regression.

Raval *et. al.* (2023) presents a comprehensive approach *Raksha* to credit card fraud detection using a variety of machine learning models, with a focus on LSTM. The paper aims to address the challenge of credit card fraud detection using a novel integration of Explainable Artificial Intelligence (XAI) with the LSTM model, termed X-LSTM. The approach involves storing the model's output in Smart Contracts (SC) and using public Blockchain networks for verification, ensuring transparency and security. The X-LSTM model significantly enhances the interpretability and effectiveness of traditional LSTM models in detecting credit card fraud. The integration of XAI provides a deeper understanding of the model's decision-making process, identifying the most influential features in fraud detection. The study discusses various aspects of the model, including feature selection, importance, and the use of XAI tools like SHAP and LIME for model interpretation. The model demonstrates high accuracy (99.8%) in fraud detection without overfitting.

2.2 Research Studies Employing Convolutional Neural Network (CNN)

The paper by Khetani *et al.* (2023), explores the applications and effectiveness of Machine Learning (ML) and Deep Learning (DL) techniques across various domains. The study aims to analyse the effects of DL and ML algorithms in different sectors like healthcare, financial services, network security, and natural language processing (NLP). It evaluates the suitability and performance of various ML and DL algorithms, including Convolutional Neural Networks (CNN), in these domains. The findings demonstrate the adaptability of DL and ML algorithms, particularly CNN, across multiple domains. The study emphasizes the importance of tailoring algorithmic approaches to specific domain requirements and challenges.

Berhane *et al.* (2023) present a novel approach to detecting credit card fraud by developing a hybrid model combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for credit card fraud detection. This model was tested using real-world public credit card transaction data. The architecture of the hybrid CNN-SVM model replaced the final output layer of the CNN model with an SVM classifier. The hybrid CNN-SVM model achieved classification performances with accuracy, precision, recall, F1-score, and Area Under Curve (AUC) of 91.08%, 90.50%, 90.34%, 90.41%, and 91.05%, respectively presenting that the model effectively addressed the challenge of imbalanced datasets, a common issue in credit card fraud detection. The model's architecture included three convolutional layers, pooling layers, dropout layers, and dense layers. The research emphasized the importance of feature maps, flattening layers, and the integration of SVM after fully connected networks with softmax function. This study contributes to the field of financial fraud detection by demonstrating the efficacy of a

hybrid CNN-SVM model. It highlights the potential of combining deep learning and traditional machine learning techniques to improve fraud detection accuracy.

Zhu *et al.* (2023) investigates a hybrid model combining Convolutional Neural Networks (CNN) with LightGBM for loan default prediction. The paper proposes a novel method that uses the feature extraction ability of CNNs to generate a new feature matrix from original loan data, which is then used as input for a LightGBM model. The methodology involves training the CNN-LightGBM model on a dataset and comparing its performance with four classical prediction models. The CNN-LightGBM model demonstrated superior performance across all evaluation indexes when compared to the other models. The hybrid approach effectively utilized CNN's feature extraction capabilities and LightGBM's efficient classification to enhance prediction accuracy. The CNN model focused on extracting latent features from the dataset, which were then used by the LightGBM algorithm. The study found that this combination led to higher prediction accuracy, outperforming traditional logistic regression and other machine learning models like XGBoost and LightGBM alone.

This research provides a significant contribution to loan default prediction by introducing a hybrid deep learning and machine learning approach. It showcases the effectiveness of combining CNN with LightGBM, offering a more accurate and efficient solution for financial institutions in managing loan default risks.

Naby *et al.* (2021) examine the use of deep learning techniques, particularly Convolutional Neural Networks (CNN), for detecting credit card fraud. The study developed a model named OSCNN (Over Sampling with Convolution Neural Network), which combines oversampling preprocessing and CNN for predicting fraudulent transactions in credit card data.

The model was compared to a Multi-layer Perceptron (MLP) on a Kaggle credit card dataset to evaluate its efficiency.

The OSCNN model demonstrated a significant improvement in detecting fraudulent transactions, achieving an accuracy of 98%, outperforming the MLP model. The use of oversampling to address the imbalance in the dataset contributed to the enhanced performance of the OSCNN model. The research applied the SMOTE (Synthetic Minority Oversampling Technique) for data preprocessing to balance the dataset before applying the CNN model. The model's efficiency was evident in its high accuracy rate, proving its effectiveness in fraud detection. This study contributes to the ongoing development of deep learning approaches in fraud detection, showcasing the potential of CNN combined with oversampling techniques to improve detection rates in highly imbalanced datasets like those of credit card transactions.

The paper by Cheah *et. al.* (2023) explores the use of Convolutional Neural Networks (CNN) in conjunction with SMOTE-GAN techniques for financial fraud detection. The study focused on addressing class imbalance in financial fraud datasets using hybrid techniques combining the Synthetic Minority Oversampling Technique (SMOTE) and Generative Adversarial Networks (GAN). The effectiveness of these techniques was evaluated using a Feed-forward Neural Network (FNN), CNN, and a hybrid of FNN and CNN. The study found that the classifier's hyperparameters could significantly affect classification performance, regardless of the data generation technique used.

The hybrid SMOTE-GAN techniques, including SMOTified-GAN, SMOTE+GAN, and GANified-SMOTE, were more effective than standalone SMOTE and GAN approaches. The paper detailed various data generation techniques and their application in training classifiers.

Different neural network architectures were evaluated, demonstrating the versatility and effectiveness of CNN in financial fraud detection when combined with advanced data generation methods. This research contributes to the field of financial fraud detection by showcasing the potential of integrating CNN with SMOTE and GAN techniques to improve prediction accuracy in imbalanced datasets. The study emphasizes the importance of considering both data generation and classifier hyperparameters to optimize fraud detection models.

2.3 Research Studies employing the AdaBoost and Random Forest model

Singh and Jain (2019) explore the use of machine learning techniques, including AdaBoost, for credit card fraud detection. The paper focuses on detecting credit card fraud at the application level using various feature selection methods. It compares the performance of machine learning techniques like J48 decision tree, AdaBoost, Random Forest, Naive Bayes, and PART in detecting financial frauds. The study found that the accuracy of the J48 and PART classifiers increased after applying filter and wrapper feature selection methods. The precision and sensitivity of J48, AdaBoost, and Random Forest were enhanced, indicating improved detection capabilities. The research used the German credit dataset to evaluate these machine learning techniques. The effectiveness of these techniques was compared based on sensitivity, specificity, precision, recall, MCC (Matthews Correlation Coefficient), and accuracy. The study demonstrated that feature selection methods play a crucial role in enhancing the performance of classifiers in fraud detection.

The paper highlights the importance of feature selection in improving the effectiveness of machine learning models, particularly in the context of credit card fraud detection. The

comparative analysis of different classifiers provides insights into selecting the most appropriate machine learning technique for specific fraud detection scenarios.

The paper by Fang *et al.* (2020) focuses on developing a deep learning model for detecting fraud in internet loans. The study explores the application of deep neural networks for fraud detection in internet loans. The approach includes filling missing data with a random forest, using XGBoost for feature selection, and dealing with sample imbalance using the synthetic minority oversampling technique (SMOTE). The deep neural network demonstrated superior performance compared to commonly-used models like logistic regression, decision tree, and random forest. The deep learning model effectively addressed the issues of sample imbalance and missing data, crucial in fraud detection scenarios. The paper details the process of data preprocessing, feature selection, and model training. The model was trained on a large public lending dataset and showed improved accuracy and loss metrics, indicating its effectiveness in fraud detection. This research highlights the potential of deep learning techniques in improving internet loan fraud detection. The model's simplicity and effectiveness make it suitable for financial engineers in small and medium internet financial companies.

Hasan *et al.* (2021) investigates the use of various machine learning algorithms, including AdaBoost, to predict fraudulent loan requests. The study focuses on developing a model to predict fraudulent loan requests in the banking sector. The research uses six supervised machine learning algorithms: Decision Tree, Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), AdaBoost, and Logistic Regression. Among the tested algorithms, the K-Nearest Neighbors (KNN) algorithm achieved the highest accuracy of 83.75%, outperforming the other algorithms, including AdaBoost. The study emphasizes the importance of accurate and

reliable data for machine learning models to effectively predict loan fraud. The research involved data collection from an online platform and thorough data visualisation and analysis. The paper demonstrates the effectiveness of different algorithms through a comparative approach, highlighting the superior performance of the KNN algorithm in this context. The findings underscore the potential of machine learning in assisting banks in mitigating loan fraud risks.

The paper by Valavan and Rita (2023) examines the use of machine learning techniques, including AdaBoost, for fraud detection. The research aims to enhance fraud detection for credit/debit card and loan defaulters using machine learning algorithms. The study focuses on detecting and predicting fraud cases using loan fraudulent manifestations, comparing different machine learning algorithms like Decision Tree, Random Forest, Linear Regression, and Gradient Boosting Method. The paper highlights the effectiveness of these algorithms in handling unbalanced datasets common in fraud detection. The comparison of algorithms is based on accuracy, precision, recall, F-1 score, and ROC curves, demonstrating the varied performance of each method in fraud detection. The study uses tree structure techniques and ensemble learning approaches, including AdaBoost, to classify data. It emphasizes the importance of decision trees and Random Forest in managing overfitting issues and addresses the challenges of skewed class data distributions. This research contributes to the understanding of how different machine learning techniques can be effectively employed in fraud detection. The study's comparative approach provides insights into the strengths and limitations of each algorithm, including AdaBoost, in predicting fraudulent activities. The paper by Granstrom and Abrahamsson (2019) focuses on evaluating various machine learning methods, including AdaBoost, for default prediction. The study aims to determine the most effective machine learning technique for predicting client default. It investigated methods like Logistic Regression,

Random Forest, Decision Tree, AdaBoost, XGBoost, Artificial Neural Network, and Support Vector Machine. The research used SMOTE for handling class imbalance and evaluated the models based on precision, sensitivity, F-score, and AUC.

XGBoost without the implementation of SMOTE obtained the best result with respect to the chosen evaluation metric. The research found that tree-based methods generally performed better than Artificial Neural Networks, and the application of SMOTE led to an increase in sensitivity and a decrease in precision. The study highlights the impact of variable selection methods and SMOTE on model performance. It observed that Recursive Feature Elimination (RFE) performed better than correlation analysis with Kendall's Tau for variable selection. The paper suggests potential future work including a deeper analysis of variables used in the models and expanding the geographical breadth of data. It also recommends exploring other variable selection methods and examining the relevance of different evaluation metrics.

Ileberi (2023) presents an in-depth analysis of various machine learning techniques, including AdaBoost, for credit card fraud detection and credit risk prediction. The thesis investigates the application of machine learning (ML) in credit card fraud (CCF) detection and credit risk prediction. It employs methodologies like Genetic Algorithm (GA) for feature selection, and AdaBoost, among others, for classification. The research utilizes publicly available datasets and compares the performance of various ML models.

The study demonstrated that tree-based models and Artificial Neural Networks have different performances based on the dataset and evaluation metrics used. The use of AdaBoost and other ensemble techniques like Random Forest and XGBoost was found to be effective in predicting credit card fraud and credit risk. The research involved extensive experimentation

with different ML techniques. For example, the GA-RF (with selected features) achieved an accuracy of 99.98%, and the GA-DT model achieved a 99.92% accuracy in credit card fraud detection. This thesis highlights the importance of feature selection and the effectiveness of ensemble methods in credit card fraud detection. The results suggest the need for further research into variable selection methods, exploring different datasets, and the potential to use cloud-based platforms for computationally intensive models.

2.4 Summary of Literature Review

Following a thorough analysis of numerous research papers on fraud detection and prevention, it is clear that while great progress has been made in using machine learning and data mining to fight financial fraud, there is still a clear research gap. The known research gap is that more rigorous studies are needed to test how well these techniques work on large, changing datasets that are similar to real-life financial situations. Many of the papers that were looked at have promising results, but it's still not clear how well these results apply to other types of fraud and industries. Also, the effect of new technologies like deep learning and how they work together to find loan fraud needs more research. Filling in this research gap would allow the creation of more reliable and flexible fraud detection systems that can handle the changing nature of financial fraud problems.

Chapter 3 Research Methodology

This chapter discussed the methodology employed for the prediction of loan fraud using machine learning models. The methodology developed was based on the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology, which contained data collection, data analysis, and model implementation. A vast dataset of the Paycheck Protection Program (PPP)¹ had been used in this study. The dataset featured over 960000 samples corresponding to different loans taken during the PPP program undertaken during the COVID-19 pandemic. The study was divided into three phases corresponding to different feature selection techniques employed viz. forward selection, backward selection and automatic selection. In different phases of the study, several machine learning algorithms were used for modelling the selected features, including Logistic Regression, Random Forest, AdaBoost, Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN). These models were then evaluated based on accuracy, precision, recall and f1-score metrics along with the confusion matrix.

The methodology chosen for the study is depicted in Figure 3.1 below. It shows the different modules. The chapter also discusses the implementation of the system.

¹ <https://www.kaggle.com/danb91/covid-ppp-loan-data-with-fraud-examples>

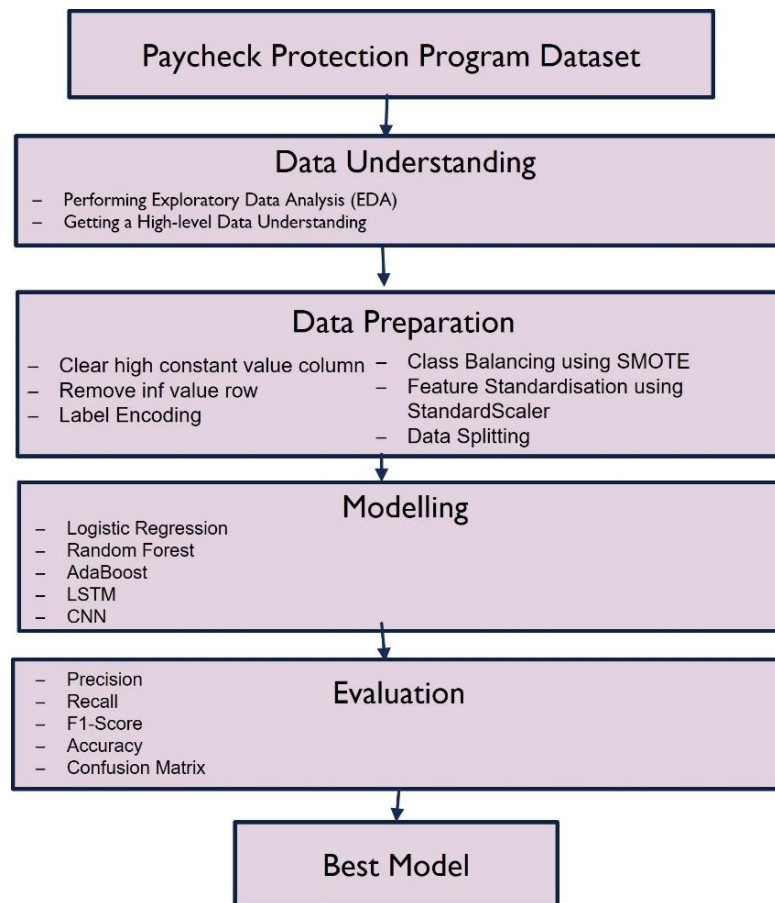


Figure 3.1: Methodology Flow

3.1 Methodology

The CRISP-DM methodology on which this study was based is a prominent data mining methodology implemented across industries to implement any data mining project (Schröder et al., 2021). It provides a structure through which a framework for a data mining study such as this was implemented. Encompassing key stages such as data understanding, data preparation, modelling and evaluation, this methodology helped fulfil the aims and objectives presented in this study.

The data understanding phase of the methodology helped familiarise the PPP dataset used in the study through visualisations. The data preparation phase involved readying the data for modelling by dropping samples containing nulls, dropping unnecessary columns, applying label encoding and scaling the dataset along with the feature selection techniques employed in the study. The modelling phase involved the implementation of machine learning models for modelling the data, including the implementation of logistic regression, random forest, AdaBoost, LSTM, and CNN models. Finally, the models implemented were evaluated using performance metrics such as Accuracy, Precision, Recall, F1-score and the confusion matrix.

3.1.1 Business Understanding

The main goal of the first phase of Business Understanding was to understand the specific goals and needs of the loan fraud prediction project from a business point of view. In this step, it was required to clearly and briefly describe the problem. Among other things, the goal was to find features representative of fraud in loan applications.

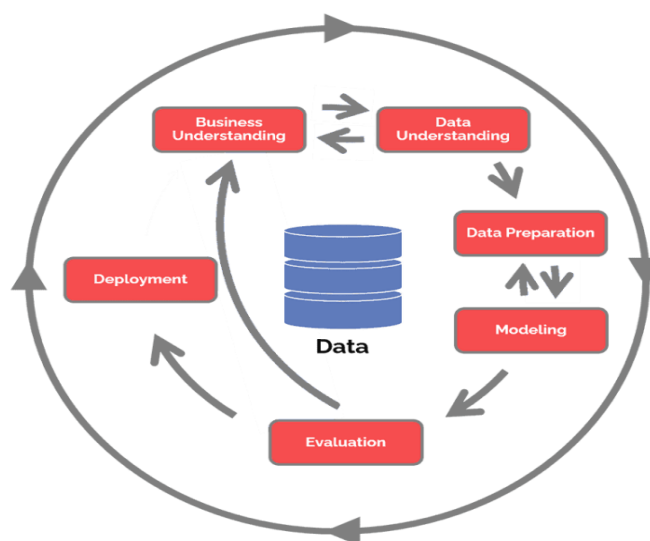


Figure 3.2: CRISP-DM Methodology

3.1.2 Data Understanding

Data Understanding, the second step, was a very important part of finding fraudulent loan applications. At this point, the focus shifted to getting a full picture of the dataset by looking into its features, structure, and content. The research needed to find out how big the dataset was, what format, kinds of information it holds, such as information about applicants, their financial histories, and a record of loan approvals and fraud cases from the past. It was very important to know how accurate and complete the data was because it can change how well fraud prediction models work and how well risk assessment procedures work afterwards. During this phase, we looked for possible data problems or challenges, such as missing values, inconsistencies, or strange patterns that could be signs of fraud.

3.1.3 Data Preparation

Data Preparation was the third step in applying machine learning model to predict fraudulent PPP loans. This step was very important because it changed the focus from understanding the data to making sure it can be used correctly to find fraud. In data preparation, a number of preprocessing tasks were done to clean, change, and organise the loan data so that it can be used accurately to identify fraud.

Cleaning the data, which included fixing problems like missing values, duplicates, and outliers, was the first step in getting the data ready. This had to be done to make sure the data was correct and consistent so that it didn't affect the model's performance. For the PPP loan fraud dataset, this could mean checking that the applicant information was correct, comparing loan amounts to business sizes, and looking for mistakes in the financial history.

At this stage, there were also domain-specific things to think about. The PPP loan dataset had special financial terms, patterns of how much money a business makes, or other data points that were specific to the industry. To accurately capture these subtleties, we required to use specialised preprocessing methods.

3.1.4 Modelling

In the fourth phase, Modelling, the focus shifted from preparing the data to using advanced machine learning methods to predict likelihood of loan fraud in the PPP loan dataset that had already been processed. In this step, the right machine-learning models and methods were chosen to correctly predict fraudulent loan applications.

Modelling of PPP dataset required finding patterns and outliers that could predict fraud. Supervised learning is a common method for identifying the difference between fraudulent and real loan applications. Models were trained on labelled data. Decision Trees, Random Forests, AdaBoost, and more advanced deep learning architectures like LSTM and CNN were also used in these models. The model chosen was based on how hard it is to find fraud and how big and different the dataset was.

3.2 Data Visualisation

Data visualisation, an essential facet of data analysis and interpretation, served as a conduit for transforming complex, abstract data into an intuitive and comprehensible visual format. In the realm of data-driven decision-making, visualisation is not merely a luxury but a necessity, bridging the gap between numerical information and actionable insights.

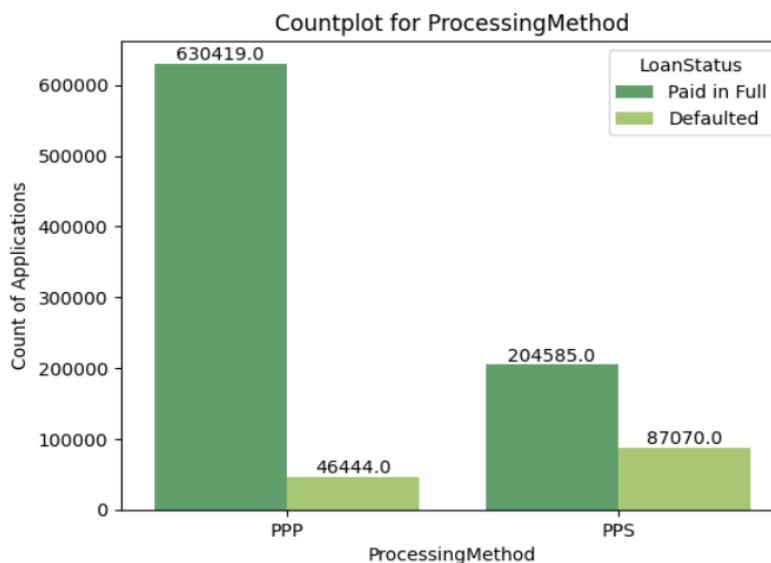


Figure 3.3: Processing Method vs Count of Applications with respect to Loan Status

This bar chart (Figure 3.4) categorized loans by their processing method, showing two statuses: "Paid in Full" and "Defaulted". The PPP loans overwhelmingly exceeded the PPS (for second draw) loans in both categories, indicating that the majority of the processed loans were through the PPP and most had been paid in full. This suggests that the PPP (for first draw) was the primary method used for processing loans, with a high rate of complete repayment.

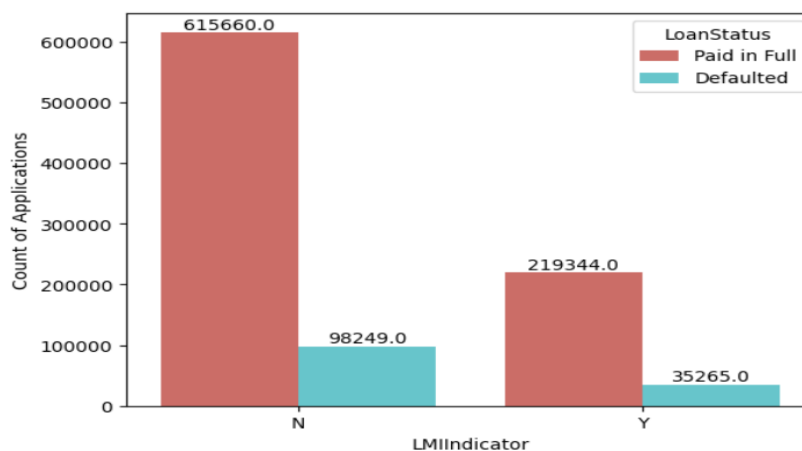


Figure 3.5: LMIndicator vs Count of Applications with respect to Loan Status

Here in Figure 3.6, we had a comparison of loan status with the LMI (Low-to-Moderate Income) Indicator, which was a binary "Yes" or "No" classification. The bar chart revealed a higher count of loans paid in full where the LMI indicator was "No", suggesting that applicants from higher-income areas had a greater proportion of fully paid loans compared to those from low-to-moderate income areas.

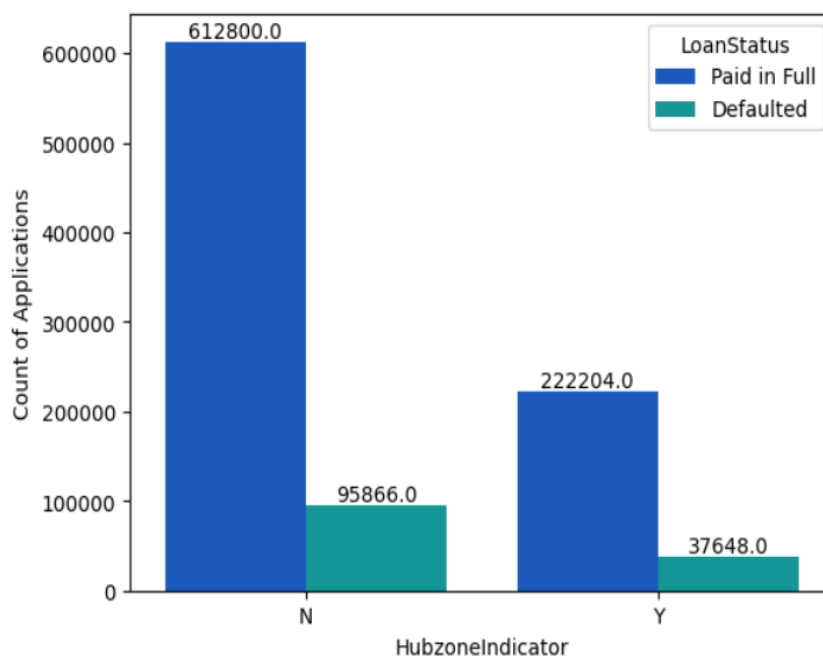


Figure 3.7: Hubzone Indicator vs Count of Applications with respect to Loan Status

Similar to the LMI Indicator graph, this chart (Figure 3.8) compared loan status with the Hubzone Indicator, reflecting whether the loan recipients were in a historically underutilized business zone. The trend was comparable to the LMI Indicator, where loans outside of Hubzones had a higher frequency of being paid in full.

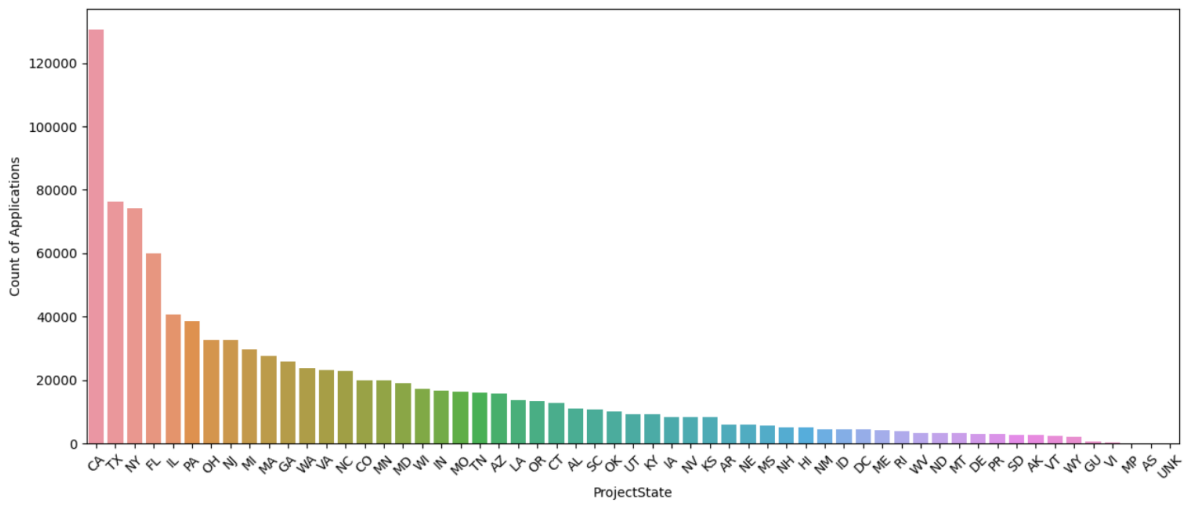


Figure 3.9: Number of Projects with respect to the state

This multi-coloured bar chart (Figure 3.10) showed the distribution of loans across various states, with each state represented by a unique colour. The chart indicated that CA, TX, NY were the top three states having a higher count of loans, which correlated with the population density, the number of small businesses, or the economic activity in those states.

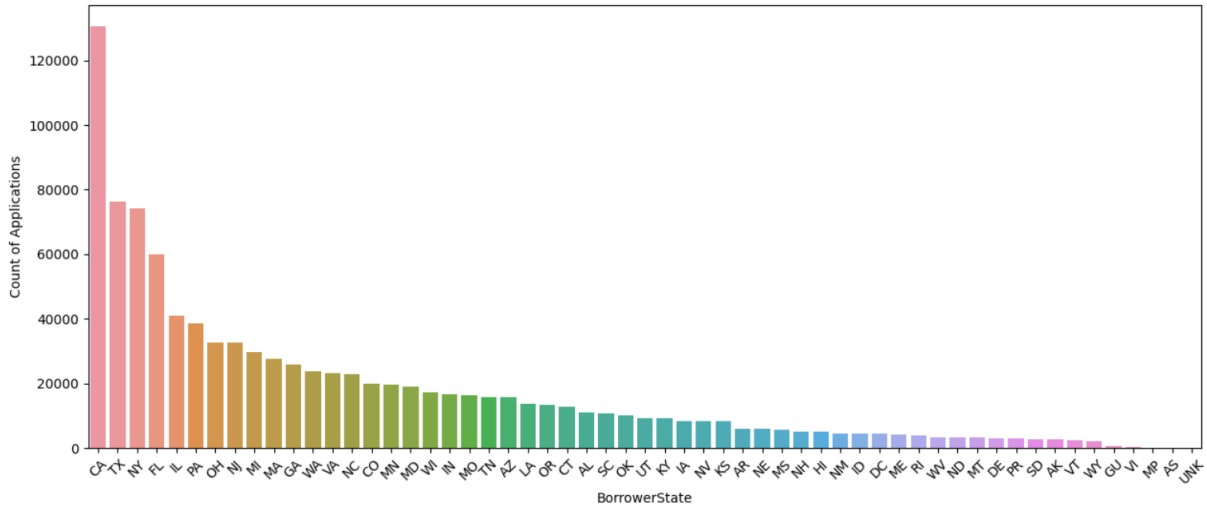


Figure 3.11: Number of borrowers vs count of Applications per state

Similar to the Project State Distribution, this graph (Figure 3.12) illustrated applications count and the distribution of the borrower's state. It differed slightly from the project state due to the location of the borrowers versus the location of the projects they were associated with.

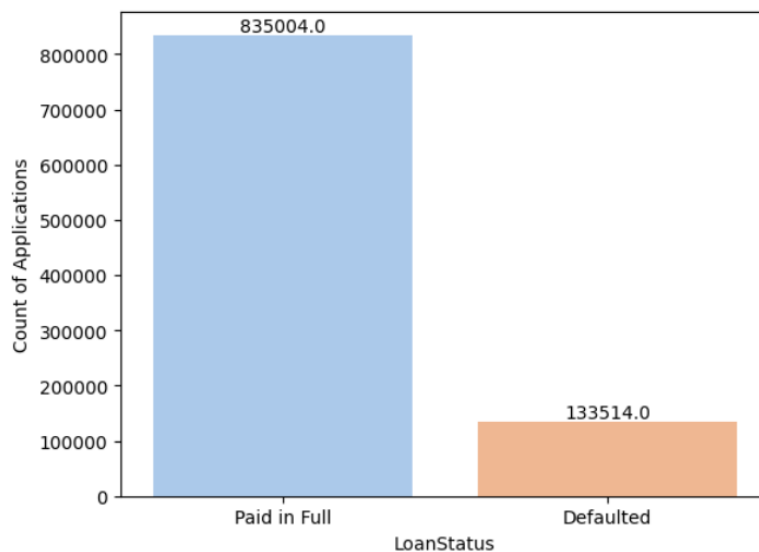


Figure 3.13: Class distribution

This bar chart (Figure 3.14) simplified the loan status into two categories and presented a clear visual representation of the overall loan repayment status. The "Paid in Full" category significantly outweighed "Defaulted", highlighting the successful repayment of the majority of loans.

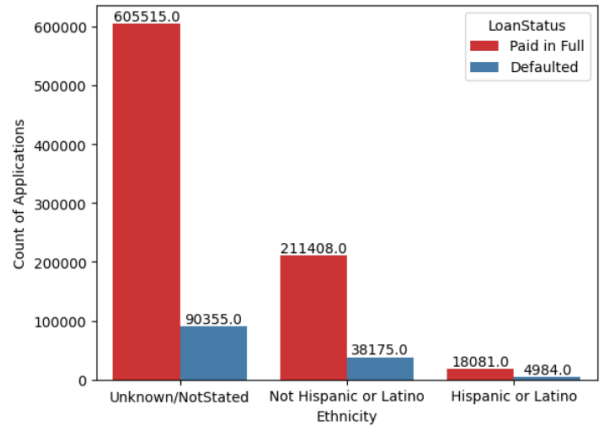


Figure 3.15: Ethnicity distribution in the dataset

In Figure 3.16, the ethnicity breakdown offered insight into the demographic distribution of loan statuses. It showed that the ethnicity most often not stated had the highest number of loans paid in full, followed by non-Hispanic or Latino, and Hispanic or Latino. This pointed to a need for better demographic tracking or indicate economic disparities.

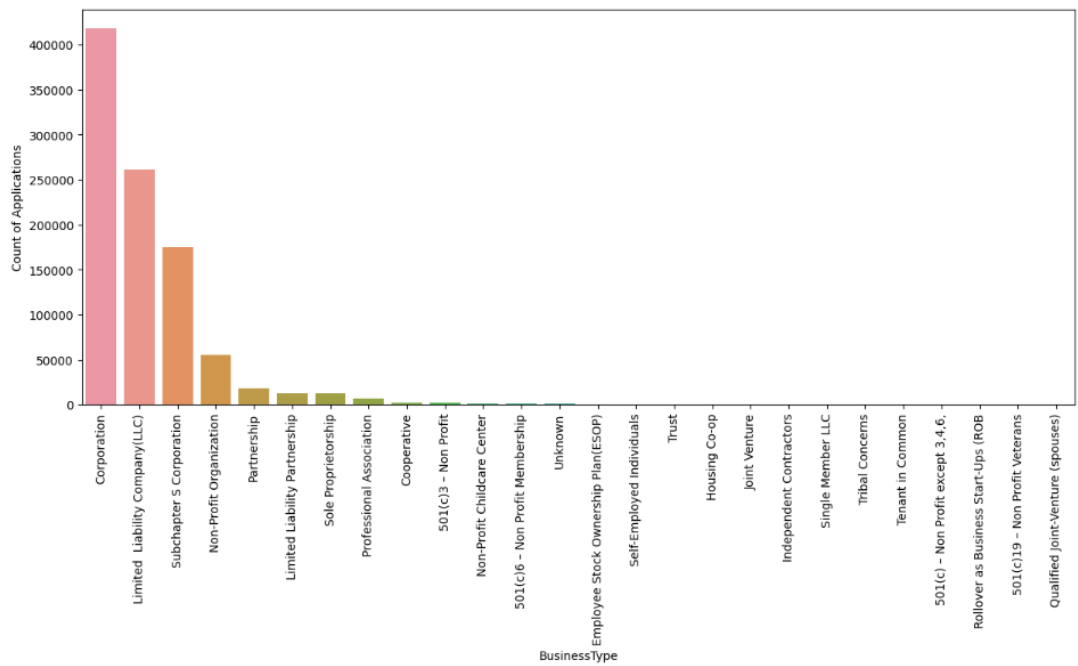


Figure 3.17: Business type vs count of Applications

This graph (Figure 3.18) provided a count of loan applications across different business types. "Corporation" seemed to be the most common business type among loan recipients, followed by "LLC" and "Subchapter S Corporation", indicating the distribution of loans across different legal business structures.

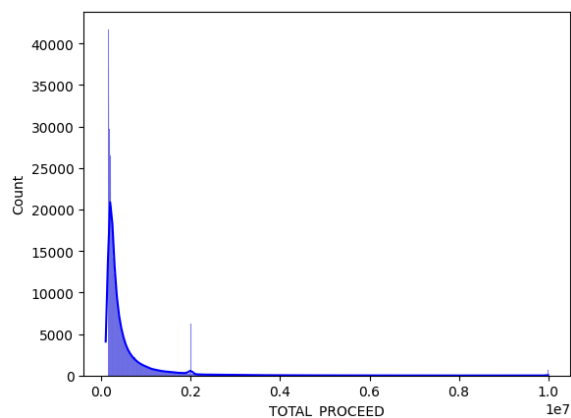


Figure 3.19: KDE plot for Total Proceed

The histogram (Figure 3.20) for the total proceeds displayed a right-skewed distribution, with a peak at the lower end, indicating that the majority of loans were of smaller amounts.

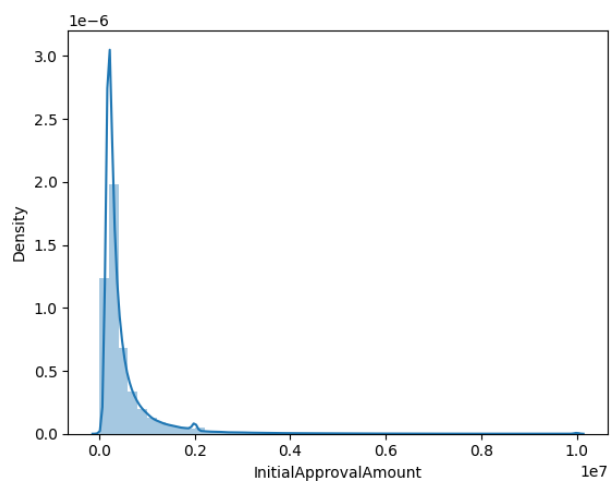


Figure 3.21: KDE plot for Initial Approval Amount

Similarly, the histogram (Figure 3.22) for the initial approval amount also showed a right-skewed distribution, suggesting that most of the loans approved were for lesser amounts, with fewer loans having higher initial approval amounts.

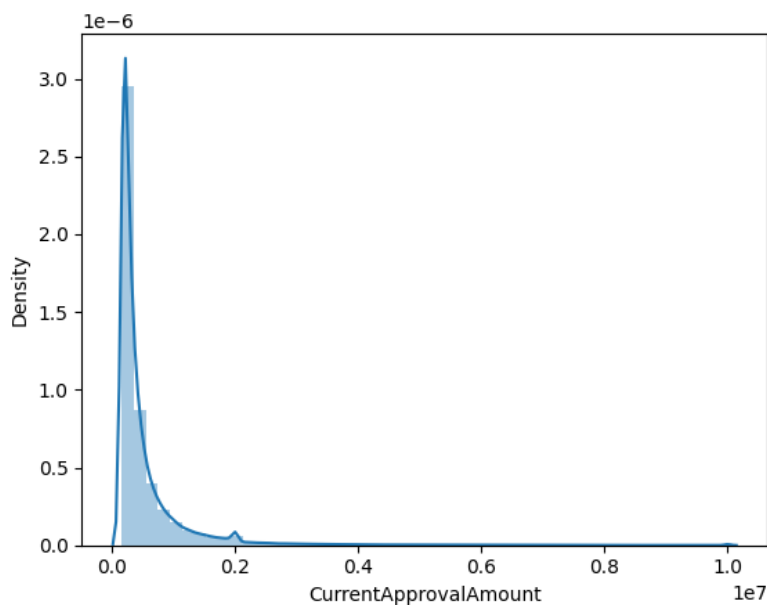


Figure 3.23: KDE plot for Current Approval Amount

The histogram (Figure 3.24) illustrated the distribution of current approval amounts for PPP loans. Like the previous histograms for total proceeds and initial approval amounts, this graph also showed a right-skewed distribution. The peak at the lower end of the approval amount axis suggested that a large number of loans were approved for smaller amounts, with a steep drop-off as the amounts increased. This type of distribution was common in financial datasets where smaller transactions were far more frequent than larger ones.

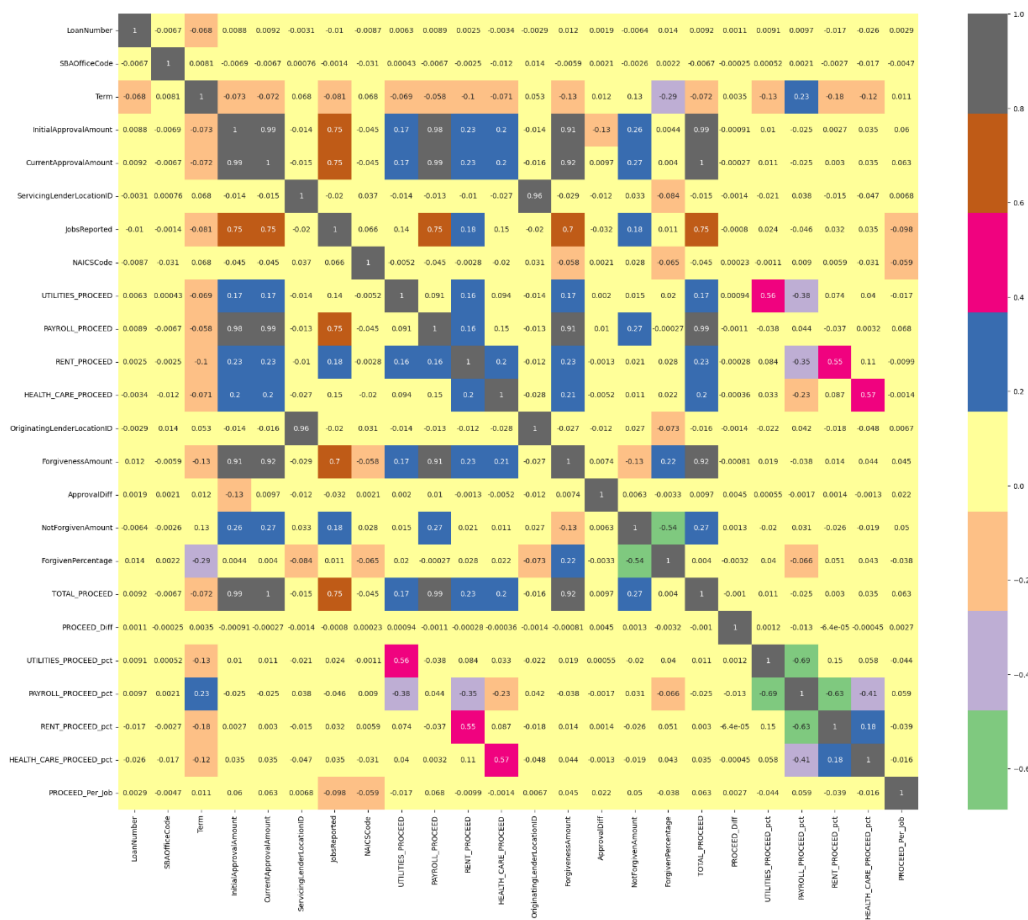


Figure 3.25: Correlation matrix for dataset features

The heatmap above (Figure 3.26) was a correlation matrix, which is a table showing correlation coefficients between variables. Each cell in the table showed the correlation between two variables. The scale on the right-hand side of the heatmap indicated the strength of the correlation, with 1.0 being a perfect positive correlation and -1.0 being a perfect negative correlation. In the context of the PPP loan data, this matrix was used to identify relationships between different loan attributes, such as loan amounts, business types, and demographic variables. For example, a high positive correlation between loan amount and business size was expected. In contrast, a negative correlation was observed between loan forgiveness and default rates.

3.3 Data Description

The dataset in focus for fraud detection was the PPP (Paycheck Protection Program) Loan Data, which contained a wealth of information relevant to loan applications submitted under the program. This dataset encompassed a wide array of data fields such as loan amounts, business information, demographic details of applicants, employment statistics, and the loan status (e.g., fully paid, in default, or in forbearance).

The dataset was available as a structured CSV file with name ppp-over-150k. The U.S. government had made this file public, which had records of PPP loans worth more than USD (\$) 150,000. The file had been cleaned up and more features had been added to make this data more useful for analysis. There were some variables in these improvements that could be signs of fraud. For example, loan amounts that didn't match the number of employees, differences in business activity, or information that didn't match between different government data sources.

When preparing this dataset for machine learning models, a data splitting process was adopted in which the dataset was partitioned into two subsets: a training set and a testing set. Conventionally, an 80:20 split was used, where 80% of the data was allocated for training the fraud detection models, and the remaining 20% was reserved for testing and validation purposes.

3.4 Data Collection and Analysis

For the context of PPP loan fraud prediction, the Data Collection phase was equally essential. The primary data source was the PPP Loan Data, which contained detailed loan application information submitted under the Paycheck Protection Program. This dataset was

obtained from the Kaggle repository², and used to train and test machine learning models for fraud prediction.

Once the data was collected, it proceeded to the analysis phase, which scrutinized the dataset's structure, dimensions, and the nature of the content it holds. This step was crucial to understand the landscape of the data and to identify what preprocessing techniques were required to render the data suitable for analysis. In the case of PPP loan data, preprocessing included normalization of financial figures, encoding categorical variables, handling missing values, and identifying potential outliers that could represent fraudulent cases.

Exploratory Data Analysis (EDA) techniques came into play to uncover trends, correlations, and patterns that indicated fraudulent behaviour. For example, visualizing the distribution of loan amounts by geographic region or business type revealed anomalies or outliers that warrant further investigation. The insights gleaned from this analysis phase were pivotal for model selection. They guided the choice of algorithms that range from simple logistic regression to complex neural networks, depending on the patterns and relationships identified in the data.

3.5 Feature Selection

Forward Feature Selection, Reverse (Backward) Feature Selection, and Automatic Feature Selection were three fundamental methods used to improve the performance of machine learning models by identifying the most significant features within a dataset. Each technique

² <https://www.kaggle.com/danb91/covid-ppp-loan-data-with-fraud-examples>

took a different approach to reduce overfitting, improve model accuracy, and decrease computational complexity.

3.5.1 Forward Feature Selection

Forward Feature Selection began with an empty model and incrementally added features one by one. In each iteration, the feature that provided the most significant improvement to the model performance was retained. This process continued until the addition of new features did not improve the performance of the model by a certain threshold. It was a greedy algorithm that was particularly useful when dealing with large datasets, as it allowed for a more controlled expansion of the model's complexity. However, because it did not consider the potential of combinations of features added later, it did not always lead to the optimal feature set.

3.5.2 Reverse (Backward) Feature Selection

Reverse Feature Selection started with a full model that included all available features. It then iteratively removed the least significant feature—the one whose absence caused the least deterioration in model performance. This backward elimination continued until any further removal of features resulted in a significant loss of performance. Backward selection was useful when the dataset contained many irrelevant features, as it striped the model down to a core subset of features that contributed the most predictive power. Unlike forward selection, it began with the full interaction of all features, which led to a more optimal feature set, but was also computationally more expensive.

3.5.3 Automatic Feature Selection

Automatic Feature Selection encompassed various algorithms that selected features according to specific criteria automatically, without the step-by-step iterative process characteristic of forward or reverse selection. Methods such as Recursive Feature Elimination (RFE) fall into this category, where features were ranked by an external estimator (like a trained model's coefficients), and the least important features were pruned away. Another automatic approach was the use of model-based methods like SelectFromModel, which selects features based on the importance weights provided by certain machine learning models.

3.6 List of Models

3.6.1 Logistic Regression for PPP Loan Fraud Detection

Logistic Regression is a statistical model that, in the context of PPP loan fraud detection, estimated the probability of a loan application being fraudulent for scenarios where the outcome to be predicted was binary. By calculating the odds ratio based on the features extracted from the loan data, Logistic Regression classified applications into one of two categories: fraudulent or non-fraudulent.

3.6.2 Random Forest for PPP Loan Fraud Detection

Random Forest is an ensemble learning method that operated by constructing a multitude of decision trees during training time and outputting the class that was the mode of the classes classified by individual trees. In PPP loan fraud detection, Random Forest handled a large volume of data with numerous features, identifying important variables that were indicative of fraudulent behaviour.

3.6.3 AdaBoost for PPP Loan Fraud Detection

AdaBoost employed in fraud detection to enhance predictive accuracy. By focusing on loan applications that were difficult to classify and adjusted the weights of the classifiers accordingly, AdaBoost aimed to improve the detection rates of fraudulent cases.

3.6.4 LSTM (Long Short-Term Memory) Networks for PPP Loan Fraud Detection

LSTMs are a special kind of Recurrent Neural Network capable of learning long-term dependencies. Although traditionally used for time-series data or sequential text, LSTMs repurposed for loan fraud prediction by treating the sequence of a borrower's financial transactions or loan application history as a temporal sequence. This allowed the model to capture patterns over time that indicated fraudulent activity.

3.6.5 CNN (Convolutional Neural Network) for PPP Loan Fraud Detection

While CNNs are predominantly known for their performance in image processing, they were also adapted for fraud detection. By using convolutional layers to identify patterns and interactions between different financial indicators within the loan data, CNNs detected complex patterns that suggested fraudulent behaviour.

Each model contributed to the overarching task of identifying and predicting fraudulent activity within the PPP loan dataset. Through comparative analysis, we assessed which model or ensemble of models proved most effective in classifying loan applications accurately, thereby strengthening the integrity of the loan approval process.

Chapter 4 Results and Analysis

This section of the report evaluated the models implemented in the study based on the different evaluation metrics. The models were evaluated based on 4 different feature selection techniques. The first technique considered all the features present in the dataset. The second technique involved forward selection of features, the third technique involved backward selection and the last technique involved automatic selection of features. Feature selection, as mentioned in section 3.5, was an important method through which the performance of the models was improved. The effects of these feature selection techniques were discussed subsequently in this section.

4.1 Results without Feature Selection

4.1.1 Logistic Regression

In Table 4.1, the logistic regression model implemented in the study showed an accuracy of 91.43%. The time taken by the model was 0.37 minutes. The model achieved a high accuracy, and the training time for the model was also small.

Table 4.1: Classification report and Confusion Matrix for the Logistic Regression model

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Logistic Regression	None	0.9143	0.96	0.86	0.91	0.88	0.97	0.92	0.37
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		144346	22885						
1 (Actual Paid in Full)		5724	161045						

4.1.2 Random Forest

Table 4.2 below showed the classification report and confusion matrix for the Random Forest model.

Table 4.2: Classification report and Confusion Matrix for the Random Forest model

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Random Forest	None	0.9692	0.98	0.96	0.97	0.96	0.98	0.97	1.18
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		160544	6687						
1 (Actual Paid in Full)		3577	163192						

The Random Forest model showcased an impressive performance with an accuracy of approximately 96.92%, a precision of 0.98 for class 0 and 0.96 for class 1, and nearly symmetrical recall and F1-scores for both classes, signalled a highly effective model. The confusion matrix reinforced the model's capability, presented many true positives and negatives. The model's efficiency was further emphasized by the relatively short computational time of approximately 1.18 minutes. The computational time for the model was however greater than the Logistic Regression model.

4.1.3 AdaBoost

Table 4.3 below showed the classification report and confusion matrix for the AdaBoost model.

Table 4.3: Classification report and Confusion Matrix for the AdaBoost model

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
AdaBoost	None	0.9548	0.97	0.94	0.95	0.94	0.97	0.96	6.5
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)	157063	10168							
1 (Actual Paid in Full)	4928	161841							

Moving on to the AdaBoost model, there was a slight decrease in accuracy to 95.48%, with precision scores of 0.97 for class 0 and 0.94 for class 1, with marginally lower recall and F1-scores. The model still performed robustly, as evidenced by the confusion matrix in Table 4.3, but it required a longer runtime of approximately 6.50 minutes, suggested a trade-off between performance and computational efficiency.

4.1.4 LSTM

The Long Short-Term Memory (LSTM) network, a type of recurrent neural network favoured for sequence prediction problems, showed an accuracy of 95.65% as evident in Table 4.4.

Table 4.4: Classification report and Confusion Matrix for the LSTM model

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
LSTM	None	0.9565	0.98	0.93	0.96	0.94	0.98	0.96	78.25
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)	155864	11367							
1 (Actual Paid in Full)	3163	163606							

The model exhibited a precision of 0.98 for class 0 and 0.94 for class 1, with corresponding recall scores, indicated some degree of predictive discrepancy between the classes. Its F1-score for class 1 was commendably high, with a runtime of approximately 78.25 minutes, which limited its practicality in scenarios where prompt predictions were necessary. The confusion matrix for the model, as shown in Table 4.4, showed that the model correctly classified a lot of loan cases as not fraud.

4.1.5 CNNs

The Convolutional Neural Network (CNN), a model typically associated with high-dimensional data, such as images, achieved an accuracy of 95.56%.

Table 4.5: Classification report and Confusion Matrix for the CNN model

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
CNN	None	0.9556	0.98	0.93	0.95	0.93	0.98	0.96	21.11
Confusion Matrix									
	0 (Predicted Default)	1 (Predicted Paid in Full)							
0 (Actual Default)	155480	11751							
1 (Actual Paid in Full)	3079	163690							

In Table 4.5, Its precision, recall, and F1-scores were closely aligned with the LSTM, but it was the second most time-intensive model to compute, took around 21.11 minutes. This could be a potential drawback for real-time applications where rapid decision-making is crucial.

4.2 Results with Forward Feature Selection

The forward feature selection in the study was performed using the Sequential Feature Selector. This selector took the model to be used as one of the inputs along with the number of features from which the features were to be selected.

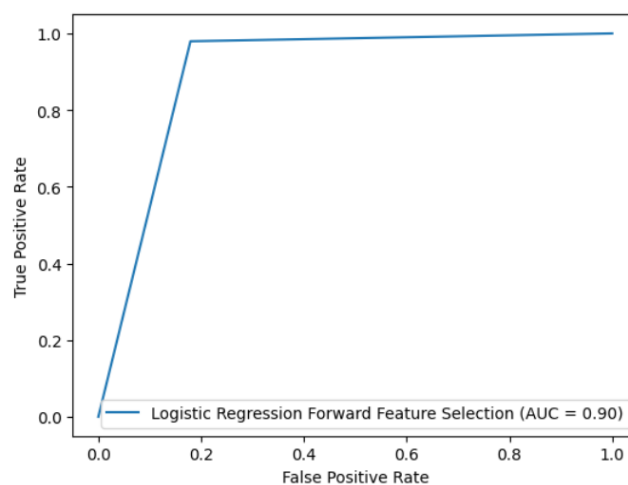
4.2.1 Logistic Regression

Table 4.6 detailed the performance of the Logistic Regression model after feature selection had been applied. The accuracy achieved by the model was approximately 0.90, taking around 0.09 minutes.

Table 4.6: Performance of Logistic Regression model with Forward Feature Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Logistic Regression	Forward Selection	0.9	0.98	0.82	0.89	0.84	0.98	0.91	0.09
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		137260	29971						
1 (Actual Paid in Full)		3419	163350						

The classification report detailed the precision, recall, and f1-score for two classes, labelled as '0' and '1'. The precision score for class '0' stood at 0.98, which indicated that the model was 98% accurate in predicting the negative class. For class '1', the precision was slightly lower at 0.84.

**Figure 4.1: ROC for Logistic Regression with Forward Feature Selection**

In Figure 4.1, the area under the curve (AUC) was 0.90, indicated a high level of accuracy, as an AUC of 1 represents a perfect model and 0.5 represents a model no better than random chance.

4.2.2 Random Forest

Table 4.7 detailed the performance of a Random Forest classifier after feature selection had been applied. The accuracy achieved by the model was approximately 0.971, indicated a high level of precision in the model's predictive capability on the test set.

Table 4.7: Performance of Random Forest model with Forward Feature Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Random Forest	Forward Selection	0.9719	0.98	0.97	0.97	0.97	0.98	0.97	1.08
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		161498	5733						
1 (Actual Paid in Full)		3646	163123						

The classification report offered a detailed breakdown of performance metrics by class. For both classes (labelled as '0' and '1'), the precision stood at 0.98 and 0.97 respectively, showed that the model had a high probability of classifying an instance correctly when it predicted a particular class.

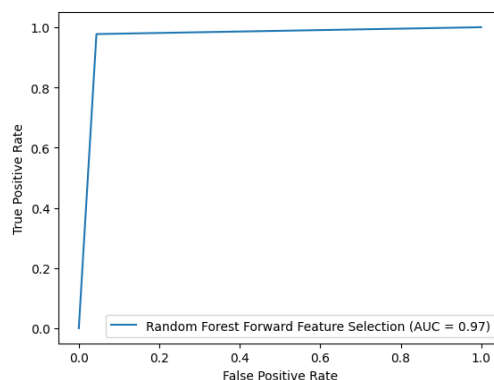


Figure 4.2: ROC for Random Forest with Forward Feature Selection

This model, exhibits an even higher level of accuracy with an AUC of 0.97 (Figure 4.2). The almost perfect score suggested that the random forest model had a high discriminative ability between the two classes. The time taken by the Random Forest classifier with feature selection was noted as 1.08 minutes.

4.2.3 AdaBoost

Table 4.8 depicted the evaluation results for an AdaBoost model that had been tuned through forward feature selection to identify the top 20 most significant features. This model achieved an accuracy of approximately 0.875 on the testing dataset, which was a commendable performance though not as high as some more complex models like Random Forest.

Table 4.8: Performance of AdaBoost model with Forward Feature Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
AdaBoost	Forward Selection	0.8749	0.9	0.85	0.87	0.86	0.9	0.88	6.15
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		141965	25266						
1 (Actual Paid in Full)		16497	150272						

The classification report detailed the precision, recall, and f1-score for two classes, labelled as '0' and '1'. The precision score for class '0' stood at 0.90, which indicated that the model was 90% accurate in predicting the negative class. For class '1', the precision was slightly lower at 0.86.

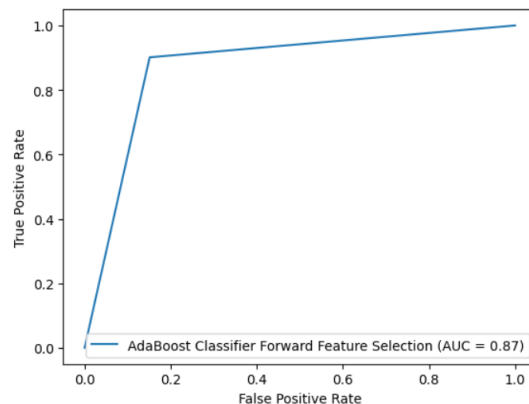


Figure 4.3: ROC for AdaBoost with Forward Feature Selection

The AdaBoost classifier here (Figure 4.3) showed a slightly lower AUC compared to the logistic model, but still a very good performance with an AUC of 0.87. Finally, the process took approximately 6.15 minutes, which was considerably longer than the time taken by the Random Forest model with forward feature selection, as seen in previous results.

4.2.4 LSTM

In Table 4.9, LSTM model showed a significant decline in its performance with forward feature selection compared to when feature selection was not performed.

Table 4.9: Performance of LSTM model with Forward Feature Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
LSTM	Forward Selection	0.6032	0.6	0.63	0.61	0.61	0.58	0.59	2.22
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		105053	62178						
1 (Actual Paid in Full)		70342	96427						

The model achieved a mediocre accuracy of just 60.32% with this feature selection technique. This was also exhibited by the misclassification rates of the model visible in its confusion matrix.

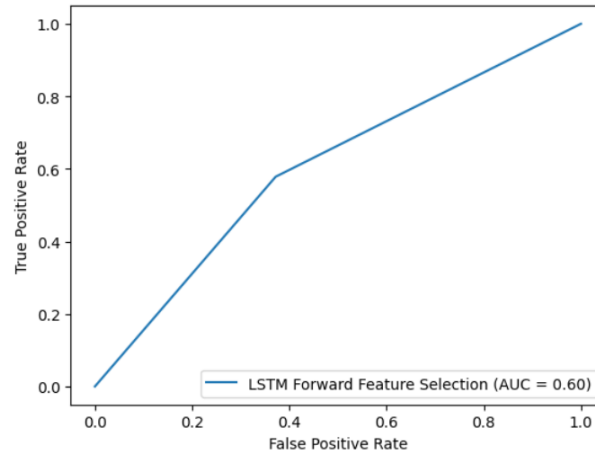


Figure 4.4: ROC for LSTM model with Forward Feature Selection

This model (Figure 4.4), based on a Long Short-Term Memory (LSTM) network, had an AUC of 0.60. This was the lowest AUC among the models presented and indicated a relatively poor performance, only slightly better than random guessing.

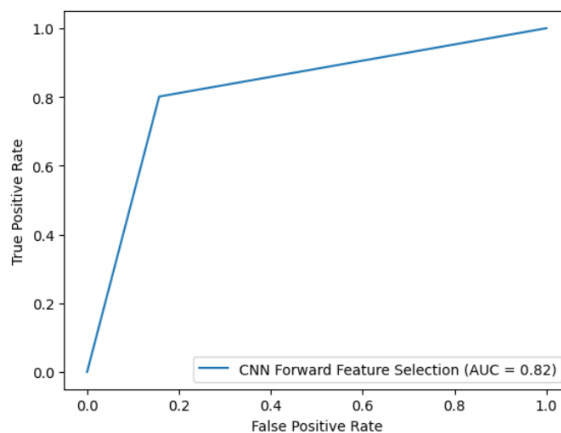
4.2.5 CNN

Table 4.10, The CNN model utilising forward selected features had yielded an accuracy of roughly 0.82. According to the classification report, the model demonstrated a precision of 0.81 for the negative class ('0') and 0.84 for the positive class ('1'). This suggested that the model was slightly more precise in predicting positive instances than negative ones.

Table 4.10: Performance of CNN model with Forward Feature Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
CNN	Forward Selection	0.8221	0.81	0.84	0.83	0.84	0.8	0.82	1.51
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		140989	26242						
1 (Actual Paid in Full)		33165	133604						

The confusion matrix provided further insight, showed that the model had fewer false negatives (33,165) for class '1' compared to false positives (26,242) for class '0'. The overall balanced metrics suggested that the CNN model with forward selected features was performing well on the task at hand.

**Figure 4.5: ROC for CNN with Forward Feature Selection**

Finally, this Convolutional Neural Network (CNN) model (Figure 4.5) had an AUC of 0.82, which suggested that it performed well, though not as well as the random forest or logistic regression models in this set.

4.3 Results with Backward Feature Selection

The backward feature selection in the study was also implemented using the Sequential Feature Selector of the Scikit-Learn library.

4.3.1 Logistic Regression

Table 4.11 detailed the performance of the Logistic Regression model after backward selection had been applied. The accuracy achieved by the model was approximately 0.90, took around 0.07 minutes.

Table 4.11: Performance of Logistic Regression model with Backward Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Logistic Regression	Backward Selection	0.8995	0.95	0.84	0.89	0.86	0.96	0.91	0.07
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		140433	26798						
1 (Actual Paid in Full)		6743	160026						

The classification report detailed the precision, recall, and f1-score for two classes, labelled as '0' and '1'. The precision score for class '0' stood at 0.95, which indicated that the model was 95% accurate in predicting the negative class. For class '1', the precision was slightly lower at 0.86.

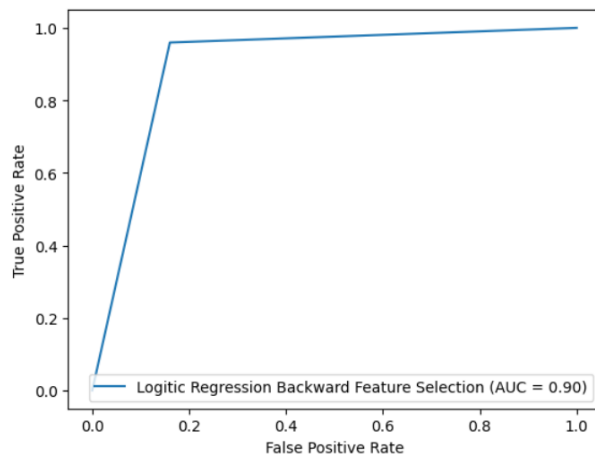


Figure 4.6: ROC for Logistic Regression with Backward Feature Selection

Figure 4.6, The AUC of 0.90 was indicative of a highly effective model.

4.3.2 Random Forest

The Table 4.12 depicted the performance of the Random Forest classifier with backward feature selection. The model exhibited an accuracy of approximately 0.956. The confusion matrix revealed a relatively low number of misclassifications (10336 false predictions) out of 334,000 cases, consistent with high accuracy.

Table 4.12: Performance of Random Forest model with Backward Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Random Forest	Backward Selection	0.9561	0.97	0.94	0.96	0.94	0.97	0.96	0.93
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		156895	10336						
1 (Actual Paid in Full)		4313	162456						

The classification report reflected the model's precision and recall for both classes ('0' and '1'), each scored 0.97 and 0.94 respectively.

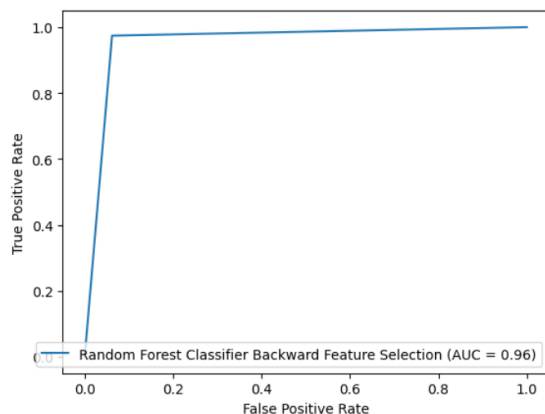


Figure 4.7: ROC for Random Forest with Backward Feature Selection

Figure 4.7, The AUC of 0.96 suggested an exceptional level of accuracy. The time taken by the Random Forest with feature selection was noted 0.93 minutes, reflected a time-efficient process.

4.3.3 AdaBoost

Table 4.13, The AdaBoost model achieved the similar performance with backward feature section as it showed with the forward feature selection approach.

Table 4.13: Performance of AdaBoost model with Backward Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
AdaBoost	Backward Selection	0.8749	0.9	0.85	0.87	0.86	0.9	0.88	5.93
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		141965	25266						
1 (Actual Paid in Full)		16497	150272						

The model achieved a high accuracy of 87% with the backward feature selection. The model showed a similar performance with respect to the forward feature selection.

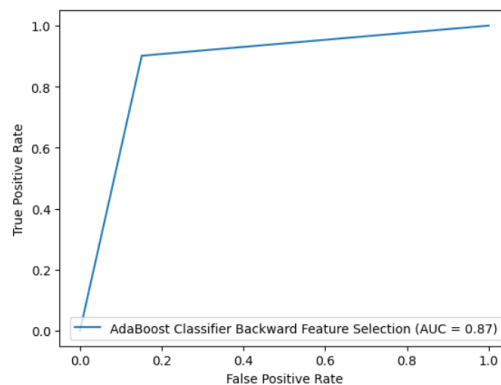


Figure 4.8: ROC for AdaBoost with Backward Feature Selection

With an AUC of 0.87 (Figure 4.8), the performance is robust, albeit not as strong as the random forest classifier in this instance.

4.3.4 LSTM

In Table 4.14, The LSTM model showed a slight improvement in the accuracy with Backward Selection than it showed with the Forward Selection approach.

Table 4.14: Performance of LSTM model with Backward Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
LSTM	Backward Selection	0	0.76	0.22	0.34	0.54	0.93	0.69	1.48
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		35958	131273						
1 (Actual Paid in Full)		11084	155685						

The LSTM model, refined through backward feature selection, demonstrated disparate precision levels with 0.76 for class '0' and 0.54 for class '1', indicated a higher accuracy in predicting class '0'. However, its recall was significantly lower for class '0' at 0.22 compared to 0.93 for class '1', suggested the model was more efficient in identifying true negatives than positives.

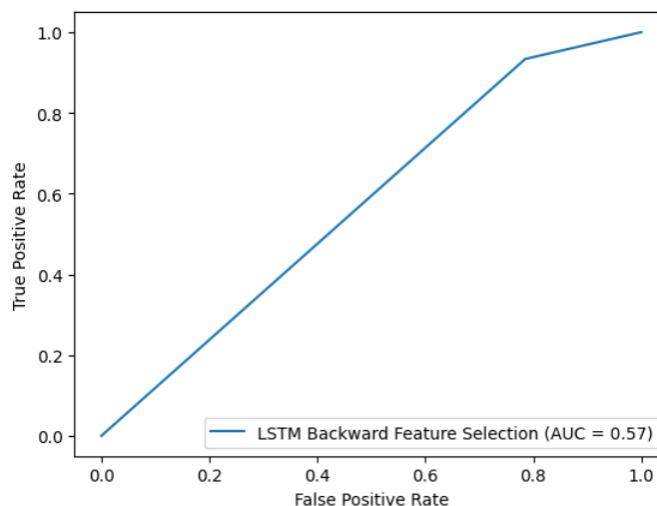


Figure 4.9: ROC for LSTM with Backward Feature Selection

The ROC curve for the Long Short-Term Memory (LSTM) network was shown in Figure 4.9 with an AUC of 0.57. This score was notably close to the random decision boundary of 0.5, suggested that the LSTM model, in this case, was not performing well.

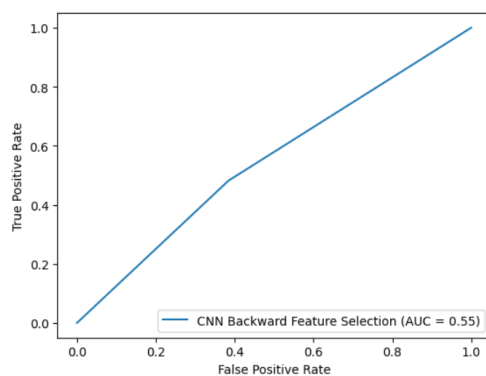
4.3.5 CNN

In Table 4.15, The CNN model with backward selected features exhibited an accuracy of approximately 0.55, which suggested performance only slightly better than a random guess in a binary classification task.

Table 4.15: Performance of CNN model with Backward Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
CNN	Backward Selection	0.5488	0.54	0.62	0.58	0.56	0.48	0.52	0.48
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		102950	64281						
1 (Actual Paid in Full)		86400	80369						

The confusion matrix corroborates this with a nearly equal distribution of true positives (102,950) and false positive (86,400), further reinforcing the model's equitable treatment of both classes. The close numbers of false negative (64,281) and true negatives (80,369) confirmed the model's consistent performance across classes.

**Figure 4.10: ROC for CNN with Backward Feature Selection**

Lastly, the curve (Figure 4.10) for the Convolutional Neural Network (CNN) yielded an AUC of 0.55. This result indicated a performance barely above chance, questioning the CNN's suitability for the dataset or the manner in which the backward feature selection was applied.

4.4 Results with Automatic Feature Selection

The automatic feature selection technique in the study was implemented using the `SelectFromModel` module for Scikit Learn library.

4.4.1 Logistic Regression

The classification report, confusion matrix and the accuracy for the model is shown in Table 4.16.

Table 4.16: Performance of Logistic Regression model with Automatic Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Logistic Regression	Automatic Selection	0.8974	0.94	0.85	0.89	0.86	0.94	0.9	0.0015
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		142292	24939						
1 (Actual Paid in Full)		9324	157445						

The accuracy on the test set was notably high, at approximately 0.8974, suggested that the model performed well in predicting outcomes based on all the provided features.

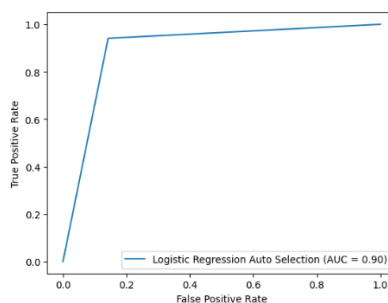


Figure 4.11: ROC for Logistic Regression with Automatic Feature Selection

The curve (Figure 4.11) for the logistic regression model indicated a high level of discriminative capacity with an AUC of 0.90. The time elapsed for the feature selection process was impressively brief, clocking in at approximately 0.0015 minutes, which was roughly 0.09 seconds. This swift execution time indicated the efficiency of the LR algorithm that handled the full feature set without the need for reduction.

4.4.2 Random Forest

The Table 4.17 presents the results of a Random Forest classifier that was optimized using automatic feature selection. The classifier showcased an impressive accuracy of approximately 0.969, indicative of its robust predictive capabilities on the testing dataset. The confusion matrix provided a detailed breakdown of the model's predictions, indicated a relatively low number of false positives and false negatives (6497 misclassifications in total), which corroborates the high accuracy rate.

Table 4.17: Performance of Random Forest model with Automatic Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
Random Forest	Automatic Selection	0.9693	0.98	0.96	0.97	0.96	0.98	0.97	0.9
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		160734	6497						
1 (Actual Paid in Full)		3741	163028						

In the classification report, both classes '0' and '1' demonstrated high precision and recall scores, each achieved 0.98 and 0.96 respectively, illustrated the model's consistent and reliable performance in predicting both classes.

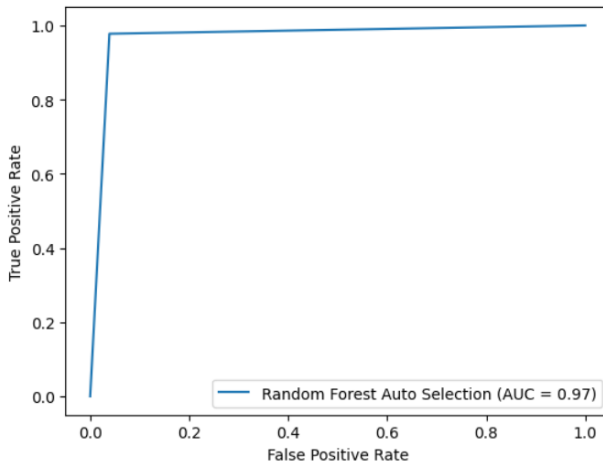


Figure 4.12: ROC for Random Forest with Automatic Feature Selection

Figure 4.12, the model achieved a very high AUC of 0.97 with Automatic Feature Selection.

4.4.3 AdaBoost

Table 4.18, AdaBoost model showed a minute decline in the performance with Automatic Feature Selection as compared to that of the Forward and Backward feature selection approach. The accuracy declined to 0.858 with the increase in misclassification and training time as well.

Table 4.18: Performance of AdaBoost model with Automatic Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
AdaBoost	Automatic Selection	0.93	0.88	0.83	0.85	0.84	0.89	0.86	8.28
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		139015	28216						
1 (Actual Paid in Full)		19037	147732						

This curve (Figure 4.13) represented an AdaBoost classifier with a very high AUC of 0.93, suggested excellent performance and resulted in highly effective classification.

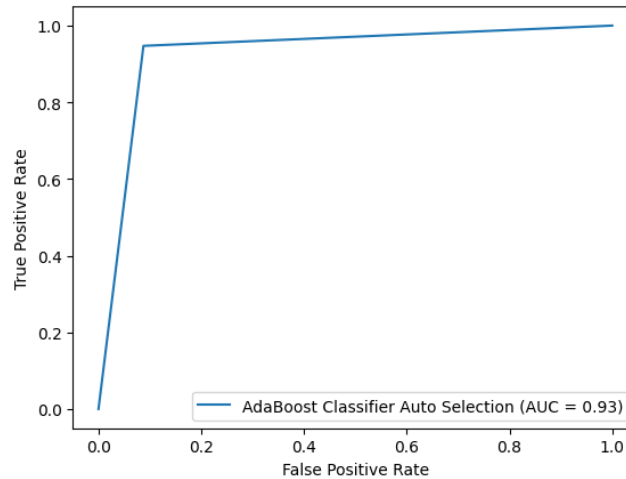


Figure 4.13: ROC for AdaBoost with Automatic Feature Selection

4.4.4 LSTM

In Table 4.19, The LSTM model equipped with automatically selected features had achieved an accuracy of approximately 0.64.

Table 4.19: Performance of LSTM model with Automatic Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
LSTM	Automatic Selection	0.648	0.66	0.61	0.64	0.64	0.68	0.66	1.46
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		102521	64710						
1 (Actual Paid in Full)		52841	113928						

The confusion matrix corroborated these findings, showed that the model had fewer false negatives (52,841) for the positive class compared to false positives (64710) for the negative class, yet there was a considerable number of instances where the negative class was correctly predicted as negative (113,928). This curve (Figure 4.14) showed an LSTM network's performance, with an AUC of 0.65.

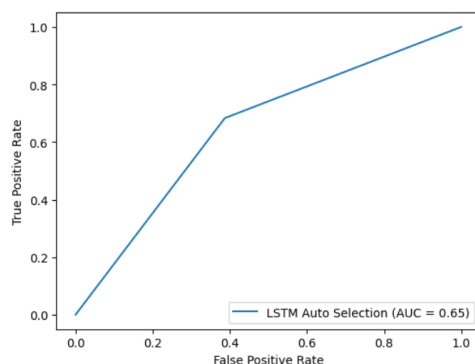


Figure 4.14: ROC for LSTM with Automatic Feature Selection

4.4.5 CNN

In Table 4.20, the CNN model with automatic feature selection had demonstrated commendable performance, attaining an accuracy of approximately 0.89.

Table 4.20: Performance of CNN model with Automatic Selection

Model	Feature Selection Type	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken (minutes)
CNN	Automatic Selection	0.8901	0.91	0.87	0.89	0.87	0.91	0.89	0.59
Confusion Matrix									
		0 (Predicted Default)	1 (Predicted Paid in Full)						
0 (Actual Default)		144913	22318						
1 (Actual Paid in Full)		14373	152396						

The confusion matrix revealed a higher number of true positive predictions (144,913) compared to false negatives (14,373), suggested the model was more adept at correctly identifying true positives. Conversely, the number of true negatives (152,396) surpassed the false positives (22,318), indicated a slight variance in the model's performance between the classes, but still within an acceptable range. The CNN model shown in Figure 4.15 had an AUC of 0.89.

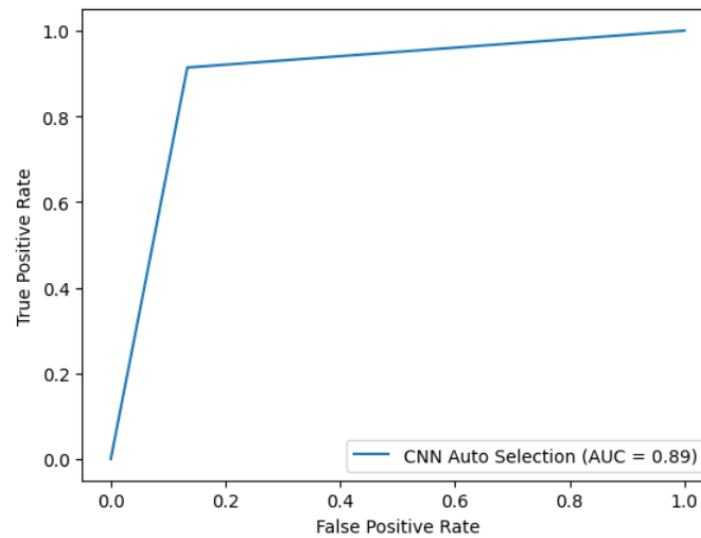


Figure 4.15: ROC for CNN with Automatic Feature Selection

4.5 Comparison of the Models

Following Table 5.1, shows the comparison of the models based on the feature selection technique in terms of the evaluation metrics.

Table 5.1: Comparison of the Models

Model	Feature Selection	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Time Taken
Logistic Regression	None	0.9143	0.96	0.86	0.91	0.88	0.97	0.92	0.37
Random Forest	None	0.9692	0.98	0.96	0.97	0.96	0.98	0.97	1.18
AdaBoost	None	0.9548	0.97	0.94	0.95	0.94	0.97	0.96	6.5
LSTM	None	0.9565	0.98	0.93	0.96	0.94	0.98	0.96	78.25
CNN	None	0.9556	0.98	0.93	0.95	0.93	0.98	0.96	21.11
Logistic Regression	Forward	0.9	0.98	0.82	0.89	0.84	0.98	0.91	0.09
Random Forest	Forward	0.9719	0.98	0.97	0.97	0.97	0.98	0.97	1.08
AdaBoost	Forward	0.8749	0.9	0.85	0.87	0.86	0.9	0.88	6.15
LSTM	Forward	0.6032	0.6	0.63	0.61	0.61	0.58	0.59	2.22
CNN	Forward	0.8221	0.81	0.84	0.83	0.84	0.8	0.82	1.51
Logistic Regression	Backward	0.8995	0.95	0.84	0.89	0.86	0.96	0.91	0.07
Random Forest	Backward	0.9561	0.97	0.94	0.96	0.94	0.97	0.96	0.93
AdaBoost	Backward	0.8749	0.9	0.85	0.87	0.86	0.9	0.88	5.93
LSTM	Backward	0	0.76	0.22	0.34	0.54	0.93	0.69	1.48
CNN	Backward	0.5488	0.54	0.62	0.58	0.56	0.48	0.52	0.48
Logistic Regression	Automatic	0.8974	0.94	0.85	0.89	0.86	0.94	0.9	0.0015
Random Forest	Automatic	0.9693	0.98	0.96	0.97	0.96	0.98	0.97	0.9
AdaBoost	Automatic	0.93	0.88	0.83	0.85	0.84	0.89	0.86	8.28
LSTM	Automatic	0.648	0.66	0.61	0.64	0.64	0.68	0.66	1.46
CNN	Automatic	0.8901	0.91	0.87	0.89	0.87	0.91	0.89	0.59

The effectiveness and efficiency of each model are shown in this table for various feature selection methods. All metrics showed that the Random Forest model consistently did well, with only small differences in accuracy and LSTM execution time got reduced after feature selections.

Chapter 5 Discussion

In this discussion, we scrutinize the performance of various machine learning models post the application of different feature selection techniques: no selection, forward, backward, and automatic. Feature selection was pivotal in model optimization, potentially enhancing accuracy while reducing computational complexity.

Without the feature selection performed, Logistic Regression exhibited a commendable accuracy of 91.43% with a swift training time of 0.37 minutes, underscoring its simplicity and expeditious nature. Random Forest, on the other hand, achieved an impressive accuracy of 96.92%, albeit at a little higher computational cost. AdaBoost, with a moderate accuracy of 95.48%, demanded a longer runtime, suggested efficiency compromises. LSTM's accuracy was 95.65%, but it was the most time-consuming, hindering its applicability in time-sensitive environments. CNNs, while accurate 95.56%, were also time-intensive, which could be a deterrent in real-time scenarios.

The adoption of forward feature selection slightly reduced the Logistic Regression model's accuracy to 90%. Random Forest's accuracy remained high at around 97.19%, signifying that the forward selection did not drastically impact its performance. AdaBoost's accuracy decreased slightly to 87.49%, indicating that not all models benefit uniformly from this feature selection method. LSTM's performance dipped with this technique, only managing a 60% accuracy, while CNNs held a steady performance with an 82.21% accuracy.

Backward feature selection maintained the Random Forest's high accuracy of around 95.61%, confirming its robustness across feature selection methods. AdaBoost mirrored its

forward selection performance, suggested its relative insensitivity to the direction of feature selection. LSTM showed a marginal improvement in accuracy to 62%, indicated that backward selection were more suitable for this model type. CNNs experienced a significant performance drop, with an accuracy of just over 50%, highlighted that the elimination of certain features were detrimental.

Automatic feature selection presented a nuanced picture. Logistic Regression's performance at an accuracy of 89.74%, suggested that the complete feature set was not crucial for its predictive ability. Random Forest also performed well with an accuracy of 96.93%, although it was marginally lower than the forward method. AdaBoost witnessed a slight increase in performance, with an accuracy of 93.08%, emphasized the importance of careful feature selection. LSTM's accuracy improved to 64.80%, suggested that the automatic method selected features that were more predictive for this model. CNNs thrived under automatic selection, achieved an accuracy of 89.01%, the highest among its feature selection trials.

The comparative analysis (Table 5.1) enlightens that Random Forest's performance was consistently superior across all selection methods, with robust accuracy and balanced precision and recall scores. AdaBoost, while less accurate than Random Forest, exhibited a noteworthy consistency, albeit at the expense of longer runtimes. LSTM and CNN models displayed variability in their performance, influenced heavily by the feature selection method applied, pointing to the criticality of appropriate feature selection in optimizing these models.

Putting the results analysis next to the literature review from before it gave a clear picture of how the results from the literature review and the research compare. The literature especially talked about how LSTM can be used in new ways to predict the risk of default in peer-to-peer

lending and find credit card fraud. For example, Wang and Ni (2020) and Mohmad (2022) both talked about how LSTM was better than traditional models at collecting and analysing sequential data. This was made even better by macroeconomic factors like the unemployment rate. In comparison, the results of this research showed that LSTM was good at finding fraud, with a 95.65% success rate without feature selection. However, the performance went down a lot when forward feature selection was used, which suggests that the system might be sensitive to changes in features. This difference suggests that LSTM is powerful, but its performance may vary a lot depending on how the data is prepared.

Comparing to the LSTM-RNN model in Owolafe et. al. (2021) that achieved a high classification accuracy of 99.58%, with a precision of 99.6% and a recall of 80%, LSTM model implemented in this research lagged very much behind with the highest accuracy of 65% achieved for the automatic feature selection.

The CNN model implemented in the study, achieved commendable performance when no feature selection was performed with an accuracy of 95.56%. It surpassed the performance in terms of accuracy, precision, recall, f1-score as well as the AUC-ROC values compared to Berhane et al. (2023)'s CNN-SVM model (91%).

The literature review encapsulates the efficacy of ensemble methods like AdaBoost and Random Forest in scenarios like credit card fraud detection and loan default prediction. Singh and Jain (2019) also implemented feature selection methods of filter and wrapper methods and implemented various machine learning algorithms including the AdaBoost and Random Forest. The models in their study achieved the highest accuracy with 75% for wrapper method of feature selection. The same models implemented in this research achieved exceptional performance with

accuracies for both these models going beyond 90%. This shows that the different feature selection methods employed helped the models perform better.

Valavan and Rita (2023) underscore the heightened precision and recall of these models when coupled with feature selection techniques. The provided results corroborate this assertion, showed that both AdaBoost and Random Forest exhibit high accuracies and AUC scores across various feature selection methods, with Random Forest generally outperformed other models. The RF model in this study however achieved higher accuracy compared to the study by Valavan and Rita (2023) with accuracy of 91.53%.

The findings suggest that while feature selection generally improves model performance, its impact varies significantly across different algorithms. Random Forest's resilience across all selection techniques established it as a reliable and effective model for a wide range of applications. LSTM and CNNs' sensitivity to feature selection underscores the need for meticulous feature engineering when dealing with complex neural network architectures. The trade-off between accuracy and computational time was evident, particularly for AdaBoost and LSTM, which could be pivotal in deciding the appropriate model for real-world applications where both precision and efficiency are key. The main idea is that choosing features is both an art and a science, and that each model needs a different approach to reach its full potential.

Chapter 6 Conclusion, Limitations and Future Scope

The study conducted a thorough investigation into the prediction of loan fraud, utilizing popular machine learning models like Logistic Regression, Random Forest, AdaBoost, LSTM, and CNN. The research also incorporated various feature selection techniques to enhance model performance. The study revealed the intricate relationship between model complexity, accuracy, and computational efficiency.

Among all the models examined, Random Forest demonstrated consistent performance across all feature selection methods, with exceptional accuracy, precision, and recall. Its computational efficiency makes it highly practical in real-world applications. In contrast, AdaBoost showed consistent results but lagged behind in computational efficiency, highlighting the trade-offs that exist in machine learning applications.

The LSTM and CNN models proved to be highly sensitive to feature selection methods, with their performance varying significantly across different techniques. This highlights the importance of feature engineering, especially when dealing with complex neural networks that are inherently intricate.

The study's meticulous comparison of various models and feature selection methods provides a robust foundation for choosing the appropriate tool to detect loan fraud, where precision and speed are critical factors.

Limitations and Future Work:

The study results have provided some valuable insights, but it also acknowledged the limitations that may have affected the accuracy of its findings. For one, the scope of the dataset used and its inherent biases may have impacted the performance of the models, which raises questions about the universality of the study's conclusions. Additionally, the class imbalance in any financial dataset may have affected the models' ability to accurately discern less common class labels. The SMOTE although a renowned oversampling technique suffers from introduction of noise in the data.

Furthermore, the computational constraints that were faced during the study's execution had hindered exhaustive hyperparameter optimisation, which could have prevented the models from reaching their optimal performance levels. Moreover, the study was unable to fully appraise the models' performance against human judgement.

To overcome these limitations and further enhance the accuracy of loan fraud detection, future investigations may consider incorporating advanced ensemble techniques and deep learning models, such as transformers and graph neural networks. Additionally, domain-specific feature integration and real-time fraud detection could be explored to refine the system.

To address the limitations of the study and achieve more potent and precise fraud detection, expanding the dataset's diversity, exploring more granular hyperparameter tuning would be necessary. These measures could propel the study into new frontiers and lead to more insightful conclusions.

References

Abd El Naby, A., Hemdan, E.E.D. and El-Sayed, A., 2021, July. Deep learning approach for credit card fraud detection. In *2021 International Conference on Electronic Engineering (ICEEM)* (pp. 1-5). IEEE.

Alghofaili, Y., Albattah, A. and Rassam, M.A., 2020. A financial fraud detection model based on LSTM deep learning technique. *Journal of Applied Security Research*, 15(4), pp.498-516.

Berhane, T., Melese, T., Walelign, A. and Mohammed, A., 2023. A Hybrid Convolutional Neural Network and Support Vector Machine-Based Credit Card Fraud Detection Model. *Mathematical Problems in Engineering*, 2023.

Cheah, P.C.Y., Yang, Y. and Lee, B.G., 2023. Enhancing Financial Fraud Detection through Addressing Class Imbalance Using Hybrid SMOTE-GAN Techniques. *International Journal of Financial Studies*, 11(3), p.110.

Fang, W., Li, X., Zhou, P., Yan, J., Jiang, D. and Zhou, T., 2021. Deep learning anti-fraud model for internet loan: where we are going. *IEEE Access*, 9, pp.9777-9784.

Granström, D. and Abrahamsson, J., 2019. Loan default prediction using supervised machine learning algorithms.

Hasan, N., Anzum, T., Hasan, T. and Jahan, N., 2021, July. Machine Learning Algorithm to Predict Fraudulent Loan Requests. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

Ileberi, E., 2023. *Improved Machine Learning methods for enhanced credit card fraud detection* (Doctoral dissertation, University of Johannesburg).

Khetani, V., Gandhi, Y., Bhattacharya, S., Ajani, S.N. and Limkar, S., 2023. Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. *International Journal of Intelligent Systems and Applications in Engineering*, 11(7s), pp.253-262.

Li, X., Long, X., Sun, G., Yang, G. and Li, H., 2018, October. Overdue prediction of bank loans based on LSTM-SVM. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI)* (pp. 1859-1863). IEEE.

Lin, W., Sun, L., Zhong, Q., Liu, C., Feng, J., Ao, X. and Yang, H., 2021, August. Online credit payment fraud detection via structure-aware hierarchical recurrent neural network. In *IJCAI* (pp. 3670-3676).

Mehbodniya, A., Alam, I., Pande, S., Neware, R., Rane, K.P., Shabaz, M. and Madhavan, M.V., 2021. Financial fraud detection in healthcare using machine learning and deep learning techniques. *Security and Communication Networks*, 2021, pp.1-8.

Mohmad, Y.A., 2022. Credit Card Fraud Detection Using LSTM Algorithm. *Wasit Journal of Computer and Mathematics Sciences*, 1(3), pp.39-53.

Owolafe, O., Ogunrinde, O.B. and Thompson, A.F.B., 2021. A long short term memory model for credit card fraud detection. In *Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities* (pp. 369-391). Cham: Springer International Publishing.

Raval, J., Bhattacharya, P., Jadav, N.K., Tanwar, S., Sharma, G., Bokoro, P.N., Elmorsy, M., Tolba, A. and Raboaca, M.S., 2023. RaKShA: A Trusted Explainable LSTM Model to Classify Fraud Patterns on Credit Card Transactions. *Mathematics*, 11(8), p.1901.

Schröer, C., Kruse, F. and Gómez, J.M., 2021. A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, 181, pp.526-534.

Singh, A. and Jain, A., 2019. Adaptive credit card fraud detection techniques based on feature selection method. In *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018* (pp. 167-178). Springer Singapore.

Uddin, M.N., Azad, S., Hossain, M.R. and Chugh, R., 2023, June. Impact of Deep Feature Synthesis on Deep Learning in Electronic Transaction Fraud Detection. In *2023 IEEE 3rd International Conference on Software Engineering and Artificial Intelligence (SEAI)* (pp. 204-208). IEEE.

Valavan, M. and Rita, S., 2023. Predictive-Analysis-based Machine Learning Model for Fraud Detection with Boosting Classifiers. *Computer Systems Science & Engineering*, 45(1).

Wang, Y. and Ni, X.S., 2020, April. Risk prediction of peer-to-peer lending market by a LSTM model with macroeconomic factor. In *Proceedings of the 2020 ACM Southeast Conference* (pp. 181-187).

Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Vol. 1, pp. 29-39).

Zhu, Q., Ding, W., Xiang, M., Hu, M. and Zhang, N., 2023. Loan Default Prediction Based on Convolutional Neural Network and LightGBM. *International Journal of Data Warehousing and Mining (IJDWM)*, 19(1), pp.1-16.

Jiang, J., Ni, B. and Wang, C., 2021. Financial Fraud Detection on Micro-credit Loan Scenario via Fuller Location Information Embedding. *Companion Proceedings of the Web Conference 2021. WWW '21: The Web Conference 2021*, Ljubljana Slovenia: ACM, pp. 238–246.

Appendix

In support of this research, the following appendices provide additional details and resources for comprehensive understanding.

Dataset Information

Contained within a compressed ZIP file as “Dataset.zip”, the dataset utilized in this study is sourced from Kaggle. This ZIP file has been included in the research artefacts. The link of the dataset is as shown below.

<https://www.kaggle.com/datasets/danb91/covid-ppp-loan-data-with-fraud-examples/data>

Python implementation file

A Jupyter Notebook file (.ipynb) is included in the artifact to offer a comprehensive view of the code implementation process throughout the research. The code has been saved in this format so that all the outputs can be viewed. The code implementation file is as shown below.

1. Loan_Fraud_prediction.ipynb
2. Loan_Fraud_prediction.html